A FRAMEWORK FOR LOW BIT-RATE SPEECH CODING IN NOISY ENVIRONMENT

A Dissertation
Presented to
The Academic Faculty

By

Venkatesh Krishnan

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
in
Electrical and Computer Engineering

School of Electrical and Computer Engineering
Georgia Institute of Technology
March 2005

A FRAMEWORK FOR LOW BIT-RATE SPEECH CODING IN NOISY ENVIRONMENT

Approved by:

Dr. David V. Anderson, Advisor
Associate Professor, School of ECE
Georgia Institute of Technology

Dr. Kwan K. Truong
Principal Engineer
Polycom Inc.

Dr. Thomas P. Barnwell III
Professor, School of ECE
Georgia Institute of Technology

Dr. Saugata Basu
Associate Professor, School of Mathematics
Georgia Institute of Technology

Dr. Mark A. Clements
Professor, School of ECE
Georgia Institute of Technology

Date Approved: March 18, 2005

To my dear parents and my teachers

ACKNOWLEDGEMENTS

At the outset, I would like to express my deep sense of gratitude to my dissertation advisor, Dr. David V. Anderson, whose guidance, support and motivation for my research led to the successful completion of this thesis. I also owe a great deal to Dr. Thomas P. Barnwell III who provided the inspiration and academic direction for the work presented in my thesis. I thank both of them for shaping my career in the digital signal processing field. I have also greatly benefited from my interactions with Dr. Kwan Truong and Dr. Mark A. Clements, and I am thankful to them for their guidance.

More than three years of graduate studies at Georgia Tech provided me with the wonderful opportunity to work closely with several colleagues and co-researchers. I cherish the fruitful interactions I had with the researchers that I worked with on the DARPA supported low bit-rate speech coding research project. I also thank all my colleagues in the Co-operative Analog and Digital Signal Processing (CADSP) group for their collaboration in several research adventures. Specifically, I cherish the moments I spent with my fellow CADSP group members, Daniel, Walter, and Heejong, during our research on hardware distributed arithmetic adaptive filter implementation, which resulted in several publications and a patent disclosure. I also sincerely thank all participants in the speech quality tests that I administered for obtaining some of the results presented in this work.

I am also thankful to my friends outside Georgia Tech for their encouragement during the past three years. Especially, I am grateful to Dimitrios and Pradeep for their moral support and their friendship. My parents mean the world to me. But for their constant encouragement and considerate support, I would not have been able to meet the challenges of this research. Last but not least, I am thankful to the Almighty for giving me the opportunity to pursue my doctoral research successfully.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

State of the art parametric speech coders, such as the mixed-excitation linear prediction (MELP) coder, employ a source–system model based representation of the speech signal. The model parameters are derived on a frame–by–frame basis from the speech signal and encoded using source coding techniques such as vector quantization. Model based coders offer a perceptually acceptable reconstructed speech quality at bit-rates as low as 2000 bits per second. However, the performance of these coders rapidly deteriorates below this rate, primarily due to the fact that very few bits are available to encode the model parameters with high fidelity. This thesis aims to meet the challenge of designing speech coders that operate at lower bit-rates while reconstructing speech at the receiver at the same or even better quality than state of the art low bit-rate speech coders. Additionally, the thesis also attempts to address the issue of designing such very low bit-rate speech coders so that they are robust to environmental noise and errors in the transmission channel.

From almost five decades of research on techniques for efficient transmission of speech over band-limited channels, it is very well known that the information that is perceptually significant varies widely from segment to segment of the speech signal. The key to the success of designing efficient very low bit-rate speech coders is in allocating the available bit resources in proportion to the perceptual significance and the information content of the features/ parameters obtained for transmission from the segments (frames) of the speech signal. For instance, if explicit classification of the frames into acoustic–phonetic units is available, then different coding models can be developed for each of these units based on their perceptual significance and the bit resources can be allocated accordingly. In one of the contributions in this thesis, we develop a plethora of techniques for efficient coding of the parameters obtained by the MELP algorithm, under the assumption that the classification of the frames

of the MELP coder is available.

Yet another class of techniques aims to allocate the bits efficiently by exploiting the correlation between the parameters/ features obtained by the speech coder from successive frames. Traditionally, encoding techniques that utilize this redundancy of information in the parameters require buffering of a large number of frames or impose structural constraints on the encoding algorithms. While the former introduces undesirable coding delays, the later renders the coding algorithms sub-optimal and therefore increases the overall distortion. In this thesis, a simple and elegant procedure, called dynamic codebook reorganization (DCR) for use in the encoders and decoders of a vector quantization system is presented that effectively exploits the correlation between vectors of parameters obtained from consecutive speech frames. The DCR procedure does not introduce any delay, distortion or sub-optimality to the encoding scheme. The potential of this technique in significantly reducing the bit-rates of speech coders is illustrated.

The rapid growth of mobile wireless communication technology requires speech coders to be designed so that they are robust to the environmental noise that corrupts the speech signal. To impart robustness, a speech enhancement framework employing Kalman filters is presented. The success of Kalman filters in many signal processing applications, including target tracking and noise cancellation is well known and its use in speech enhancement has also been reported in the past. Kalman filters designed for speech enhancement in the presence of noise assume an autoregressive model for the speech signal. We improve the performance of Kalman filters in speech enhancement by constraining the parameters of the autoregressive models to belong to a codebook trained on clean speech. We then extend this formulation to the design of a novel framework, called the multiple input Kalman filter (MIKF), that optimally combines the outputs from several speech enhancement systems. We demonstrate that the fusion of the outputs of speech enhancement systems by the MIKF yields an improved

estimate of the clean speech signal.

Since the low bit-rate speech coders compress the parameters significantly, it is very important to protect the transmitted information from errors in the communication channel. Although appropriate error correction codes may be employed to undo the effects of such noise, it is often desirable to mitigate the effects without increasing the transmitted bit-rates. For this purpose, channel-optimized vector quantizers will be used. These vector quantizers are designed by optimizing the codebook for both the source and the communication channel characteristics. In this thesis, a novel channel-optimized multi-stage vector quantization (CO-MSVQ) codec is presented, in which the stage codebooks are jointly designed. The proposed codec uses a source and channel-dependent distortion measure to encode line spectral frequencies derived from segments of the speech signal. Extensive simulation results are provided to demonstrate the consistent reduction in both the mean and the variance of the spectral distortion obtained using the proposed codec relative to the conventional sub-optimal channel matched-MSVQ.

CHAPTER 1

INTRODUCTION

Over the past fifty years, significant advances have been made in human speech communications technology. In particular, wireless voice communication systems have seen a world-wide growth in the past decade. With the increase in number of users of these technologies, the cutting edge research is now focused on designing speech processing methods that enable the design of evermore bandwidth efficient, higher quality and secure voice communication systems. The research in this field has drawn richly from advances in related areas such as speech enhancement in the presence of noise, human language understanding, human computer interaction, and voice modification. State of the art speech coding systems typically consist of a speech enhancement front-end that improves the performance of the coder in noisy environments, an efficient speech compression algorithm that aims to represent the speech signal with as few bits as possible, and a scheme to mitigate the effects of channel errors in the transmitted parameters.

The explosive growth of the human speech communications technology has been largely enabled by the digital representation and processing of speech signals. Direct digital representation of the sampled speech signal has a large amount of redundancy and is not suitable for a communication system that is constrained by bandwidth limitations. For instance, a speech signal sampled at 8000 samples per second, with each sample represented by a 16 bit digital codeword, would require transmission of 128 K bits per second. This would require a communications bandwidth that is not practical by today's standards for wireless communication systems. To analyze the redundancy in such a representation, let us assume that language being spoken is English, which has approximately 40 phonemes. Each of these 40 phonemes can be represented using a unique six bit codeword. If we assume that the average human

utters about six phonemes every second the information contained in the speech signal could be transmitted using 36 bits per second! Thus the digitized speech signal can be compressed to a large extent. Modern day speech coders operate at bit-rates as low as 2000 bits per second, while maintaining the intelligibility and perceptual quality of the reconstructed speech signal almost as good as the original.

Personal and mobile communication systems use digital multiple access techniques such as TDMA and CDMA. The use of multiple access techniques have evoked a keen research interest in variable rate speech coders, which allow a flexible allocation of bits to different segments of the speech signal. By dynamically allocating the bits to different regions of the speech signal based on the information content of that region, variable rate speech coders allow the communication system to use the available bandwidth more efficiently.

Speech signals collected using acoustic sensors, such as microphones, in noisy environments deteriorate the quality of speech in a communication system. For example, speech signals in mobile telephone systems are usually corrupted by background noise generated by the car engine, fans, etc. Likewise, in air–ground communication systems, cockpit noise corrupts the pilot's speech. Transmission of such noisy signals over the communication channel often results in severely impaired intelligibility at the receiver. In order to improve the performance of speech communication systems in noisy environments, it is important to employ efficient speech enhancement algorithms prior to transmission of the signal.

In a communication system, the transmitted signals may also be affected by noise in the communication channel. The effects of such noise is manifested as errors in the received digital codewords. In a system where the actual signal has been compressed to a large extent, such errors can cause catastrophic degradation in the quality of the reconstructed signal. Therefore, for such systems, it is important to protect the transmitted information. Efficient error protection can be provided at the cost of an

increase in the number of transmitted bits. Additionally, joint source–channel coding techniques may be employed to mitigate the effects of such channel errors, without any increase in the transmitted bit-rate.

1.1 Contributions of the thesis

In this thesis, a framework for speech enhancement, low bit-rate coding and channel error protection is presented. One of the key contributions of this thesis is the low bit-rate speech coding paradigm employing vector quantization with dynamic codebook re-ordering. This technique has the potential to dramatically reduce the bit-rates in speech coders, without introducing any additional coding delays or distortions. The main contributions of this thesis include:

1. An improvement in the performance of Kalman filters used in speech enhancement by constraining the autoregressive model parameters used by the Kalman filter to belong to a codebook trained on clean speech,

2. A novel multiple input Kalman filtering framework that allows the fusion of outputs from multiple speech enhancement systems to obtain an improved estimate of the clean speech signal,

3. A range of modifications to the MELP vocoder to reduce the transmitted bit-rates and improve the quality of the reconstructed speech, given externally supplied information about the classification of frames into different acoustic–phonetic units,

4. A novel dynamic codebook re-ordering algorithm for use in vector quantization encoder and decoder that significantly reduces the number of bits required to code vectors of parameters derived from consecutive frames of a correlated signal,

5. Application of the dynamic codebook re-ordering procedure to the MELP speech coding algorithm and demonstration of the potential for significant reduction in the bit-rate required for transmission of MELP parameters, and

6. Joint design procedure for channel-optimized multi-stage vector quantizers, that mitigate the effect of channel errors on the vector quantization indices better than the traditional sequential design procedure.

1.2 Organization of the thesis

The following chapter provides a detailed survey of some of the significant milestones in the past five decades of research on speech enhancement, low bit-rate speech coding and joint source–channel coding systems.

In Chapter 3, the Kalman filter based speech enhancement system with codebook constrained estimation of the autoregressive model parameters is described. Extensions to this formulation that allow the outputs from several speech enhancement systems to be combined to provide an improved estimate of clean speech is developed in Chapter 4. In Chapter 5, enhancements to the MELP coder algorithm and the reduction in bit-rates achievable when the classification of the speech frames into various acoustic–phonetic classes is described. The dynamic codebook re-ordering procedure is introduced in Chapter 6 and the significant coding gains achievable when this technique used in the vector quantization of Gauss–Markov vector sources is demonstrated. This technique is applied to encoding the parameters of the MELP coder in Chapter 7. The joint codebook design algorithm for channel-optimized multi-stage vector quantizer is discussed in Chapter 8.

CHAPTER 2
BACKGROUND

Over the past three decades, significant advances in the digital speech processing research has catalyzed the explosive growth of wired and wireless communication systems. Some of the key developments that have enabled this revolution include: digital representation of signals so that they can be processed on a digital computer, a plethora of lossless and lossy signal compression algorithms, linear prediction model based representation of the speech signal and the techniques for robust communication of information over wired and wireless channels. Further, several speech enhancement techniques have been developed that improve the quality of the voice communication systems in environments which are noisy.

Typically, a modern voice communication system consists of a noise pre-processor, a speech codec and a channel error protection scheme. The speech input to a practical voice communication system is often corrupted by the noise in the environment of the user of the system. The noise preprocessor aims to mitigate the effect of this noise on the quality of the speech signal that is transmitted. The speech coder is the key component in a voice communication system. The codec consists of an encoder at the transmitting side that aims to represent the information in the signal efficiently. The decoder at the receiving end reconstructs the speech signal from the information that it receives from the transmitter. An efficient speech codec (encoder–decoder pair) attempts to ensure that the quality of reconstructed speech is perceptually similar to the original speech for a given amount of information that can be transmitted over the communication channel. Channel error protection schemes ensure that the transmission of information over the communication channel is robust. In this chapter, an extensive survey of the state of the art speech coding, speech enhancement, and channel error protection algorithms is provided.

2.1 Digital representation of speech signals

In general speech coding may be defined as a process that generates sequences of binary digits from the speech signal. The goal of the modern-day speech coders is to devise a compact digital representation of the speech signal that will enable their transmission through band-limited channels and yield a perceptually acceptable reconstruction at the receiver.

Speech signal sampled at 8000 Hz and each sample quantized to 8 bits results in a data-rate of 64 Kbps. The quality of the speech thus represented is is indistinguishable from the 4 KHz band-limited analog speech and often referred to as *broadcast* quality speech [86]. This sampled and quantized representation of the speech is typically obtained at the output of analog to digital converters and forms the primary signal input to all speech processing algorithms.

The sampled and quantized speech signal (henceforth just referred to as the original speech signal) is seldom transmitted directly at 64 Kbps. A significant reduction in transmitted data-rate at the cost of a marginal degradation in the perceptual quality of the reconstructed speech can be achieved by employing a speech coder. Speech coders may broadly be classified into three groups: *waveform coders*, *parametric coders* (or model based) and *hybrid coders* [86]. Waveform coders seek to represent the waveform of the speech signal using digital symbols and typically operate at bit-rates in the 16–64 Kbps range. To achieve speech coding at bit-rates below 16 Kbps, a source–system model [18] for the generation of the speech signals is often assumed. Such a model seeks to capture the perceptually significant information in a speech signal by modeling it as the output of an autoregressive system whose input is an excitation signal inspired by the mechanism of generation of speech by humans. Fully parametric model based coders compress the speech signal by efficiently encoding the parameters of the autoregressive model and the source for speech generation. At the decoder, the received parameters are used to reconstruct the speech generation model,

which is then used to synthesize the speech signal. The efficiency of such a speech coding system largely depends on the success of the model in representing the perceptually important components of the speech signal. While fully parametric coders can achieve speech compression to bit-rates below 8 Kbps, the quality of the synthesized speech is often poor due to deficiencies in the models. The hybrid coders seek to balance the tradeoff between the bit-rate and the speech reconstruction quality. While hybrid coders work within the paradigm of the source filter model, they still try to match the speech signal waveform to the output of the speech model. Typically this is done by performing analysis of the speech signal to estimate the model parameters via synthesis. Such coders encode speech at bit-rates in the 8–16 Kbps and achieve a quality similar to that of waveform coders

In the following subsections, we briefly review speech coders that belong to the three categories discussed above. We also describe, in detail, one of the most useful coding techniques, viz vector quantization and its implementations for encoding the parameters of the speech model or the speech waveform.

2.1.1 Wavefrorm coders

Speech waveform coders transmit information to the decoder that enables the reconstruction of the speech signal waveform at the receiver end. In designing waveform coders, the goal is to minimize the distortion in the reconstructed waveform with respect to the original speech waveform, given a channel capacity that allows a limited transmission of information. The distortion is measured in terms of the mean squared error between the original and the reconstructed speech signal and the channel capacity is defined in terms of the bandwidth available for the transmission of information.

Several waveform coders exist that transmit speech waveform information in the 16–64 Kbps data-rate range and the reconstructed speech is broadcast quality. To reduce the data-rate from the original 64 Kbps, most waveform coders exploit the

correlation between successive samples in the speech waveform. Due to the presence of this correlation, the speech sample at a given time instance can be predicted as a suitably designed linear combination of a finite number of past sample values. The prediction procedure is termed linear prediction and forms the basis of several waveform coders including differential pulse code modulation (DPCM), delta modulation (DM) and adaptive differential pulse code modulation (ADPCM). Details of these coding schemes can be found in [86], [16], [37], [38]. The ADPCM was also adopted in 1991 as the G.726 ITU-T standard for speech waveform coding. [47].

Sub-band and transform coders are waveform coders that exploit the redundancies in the transform domain representation of the speech waveform. A frame of speech samples is first represented in the transform domain using either a bank of filters (sub-band coder [12] [14] [15]) or via unitary transforms (transform coders [8] [13]).

2.1.2 Model based speech coders

Waveform coders described in Section 2.1.1 are designed so that the speech waveform can be communicated over the channel with high fidelity and thus are very general and applicable to other signal types besides speech. Furthermore, the quality of the reconstructed speech signal degrades drastically if waveform coders are employed at bit-rates below 16 Kbps.

It is well known from several years of research on speech properties that it is possible to have two speech signals that are perceptually indistinguishable while their waveforms do not match. Therefore it would be useful to device a model that would represent the perceptually significant information in the speech signal. It has been found that the autoregressive (AR) model for speech describes the process of generation of the speech signals by humans and the parameters of this model contain the perceptually significant information in the speech signal. Other successful speech models include the sinusoidal model based vocoder introduced in [72] and multiband excitation vocoder [40].

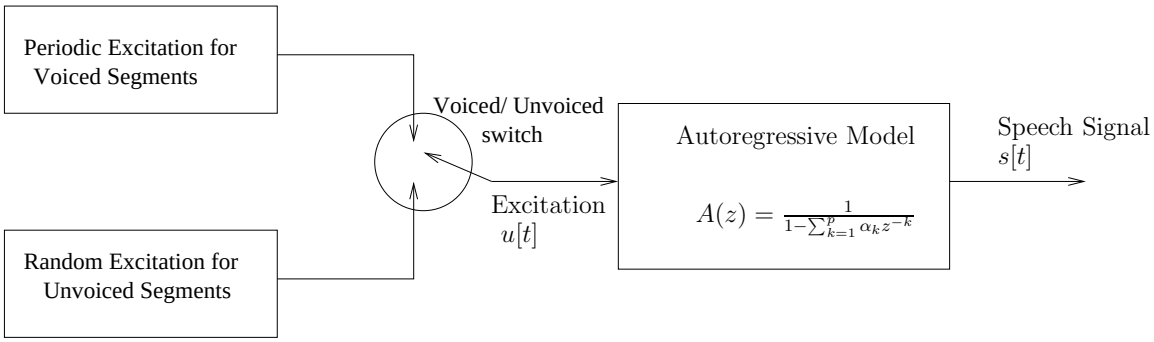

Figure 1. Autoregressive model of speech generation

A brief discussion on the AR model is provided below. Speech signal is considered to be the output of an autoregressive (AR) system, as shown in Fig. 1. The parameters of the AR model are obtained by short-time linear prediction (LP) analysis . A detailed tutorial review of linear prediction is provided in [67]. The input (also called the excitation) to this system is a periodic source signal during voiced segments of speech and a random, white noise like signal during the unvoiced segments. A p^{th} order AR model, whose parameters are $\alpha_1, \alpha_2, \ldots, \alpha_p$, is given by the Z domain transfer function:

$$H(z) = \frac{1}{A(z)} = \frac{1}{1 - \sum_{k=1}^{p} \alpha_k z^{-k}}. \tag{1}$$

If $s[t]$ is a sample of speech at t and $u[t]$ is the excitation signal, then from (1),

$$s[t] = \sum_{k=1}^{p} \alpha_k s[t-k] + u[t] \tag{2}$$

The LP analysis is the process of obtaining the AR model parameters α_k's from the speech signal. From the statistical analysis of the speech signal, it is known that speech is quasi-stationary, i.e., stationary for short periods of time (10-40 msec). Thus the AR model parameters can be assumed to be stationary for that period of time. For a short duration (say $t = [0, T]$) stationary frame of the speech signal, the AR parameters can be obtained by minimizing the mean squared prediction error over a finite window of the speech signal. If a windowed speech frame of duration T

samples and starting at a sample instance τ is

$$s_\tau[t] = \begin{cases} s[t-\tau], & t = \tau, \ldots, T+\tau \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

then minimization of the mean square prediction equation results in the Yule–Walker equation [80]:

$$\sum_{t=-\infty}^{\infty} s_\tau[t-i]s_\tau[t] = \sum_{k=1}^{p} \alpha_k \sum_{t=-\infty}^{\infty} s_\tau[t-i]s_\tau[t-k], 1 \leq i \leq p \tag{4}$$

These equations may be solved using the Levinson–Durbin recursion [80] [44]. The LP analysis filter, $A(z)$ de-correlates the excitation and the impulse response of the all-pole synthesis filter to generate the prediction residual that is an estimate of the excitation signal.

Parametric speech coders employ vector or scalar quantization [35] for encoding the parameters of the LP model and the excitation signal. Since the zeros of the LP analysis filter are poles of the AR synthesis filter, it is important that the quantized LP filter still has its zeros inside the unit circle. In other words, the quantization process must not cause the zeros of the LP analysis filter to move outside the unit circle in the z plane. This can be ensured by converting the LP parameters α_k's into a set of line spectral frequencies (LSFs) [45]. The LSFs are the roots of a symmetric polynomial $P(z)$ and an antisymmetric polynomial $Q(z)$ formed from the LP filter $A(z)$ as

$$P(z) = A(z) + z^{-(p+1)}A(z^{-1}) \tag{5}$$

$$Q(z) = A(z) - z^{-(p+1)}A(z^{-1}) \tag{6}$$

Rewriting the above equation (5) and (6), we have

$$P(z) = (1+z^{-1}) \prod_{j=1}^{p} \left(1 - 2\lambda_{2j-1}z^{-1} + z^{-2}\right) \tag{7}$$

$$Q(z) = (1-z^{-1}) \prod_{j=1}^{p} \left(1 - 2\lambda_{2j}z^{-1} + z^{-2}\right) \tag{8}$$

The p conjugate roots, $\lambda_0, \lambda_1, \ldots, \lambda_{p-1}$ of the above polynomials are called the line spectral pairs. It can be shown that if all zeros of $A(z)$ lie within the unit circle, then $|\lambda_j| \leq 1$, for all $j = 0, 1, \ldots, p-1$ [85]. If we set

$$\lambda_j = \cos(\mathbf{x}_j), \tag{9}$$

then the set $\mathbf{x}_j$, for $j = 0, 1, \ldots, p-1$ is called the line spectral frequency (LSFs). The LSFs can be obtained from the LP coefficients by a reversible procedure. Also, for a LP analysis filter that has all its zeros within the unit circle, it can be shown [82] [85] that the LSFs are arranged as

$$\mathbf{x}_0 < \mathbf{x}_1 < \mathbf{x}_2 \ldots \mathbf{x}_{p-1} \tag{10}$$

If this order is still maintained after the quantization process, the the synthesis filter reconstructed at the receiver is guaranteed to be stable.

In the following two subsections, fully parametric speech coders and the hybrid coders are briefly reviewed.

2.1.2.1 Fully parametric speech coders

Parametric coders represent the parameters of the LP model using an appropriate coding scheme such as scalar/ vector quantization. While the waveforms of the reconstructed speech may be very different from that of the original speech signal, they sound almost the same. Since the parameters of the speech models can be very efficiently encoded, fully parametric speech coders can efficiently operate at bit-rates below 8 kbps.

In the past, several model based speech coders have been proposed including the channel vocoder and the formant vocoder [18]. Popular among parametric coders are the LPC vocoders that not only encode the LP parameters using a scalar or vector quantization scheme but also represent the excitation signal parametrically. To parametrically represent the excitation signal, a given speech frame is first classified as voiced or unvoiced. The excitation signal corresponding to a voiced frame

is parameterized by the periodicity of the prediction residual. At the decoder the voiced excitation is synthesized as a train of periodic impulses and the unvoiced excitation is simply reconstructed as a white noise signal. The excitation signal is then filtered using the LP synthesis filter built from the received AR model parameters to reconstruct the speech signal.

Deficiencies in the representation of the excitation signal as a pulse train or a random noise results in poor perceptual quality of the reconstructed speech. The synthetic speech often sounds mechanical, and has a buzzy quality. Furthermore, the misclassification of the speech signal results in thumps for unvoiced speech and whispered quality for voiced speech signals. Because of its simplicity, the algorithm is not robust to background noise in harsh environments. However since only a very few parameters are transmitted in this scheme, the simple LPC model can encode speech at bit-rates as low as 2400 bps. The LPC-10 speech coder based on this model was adopted as the Federal Standard speech coder in 1984 [90].

A much improved fully parametric coder employing an improved representation of the excitation signal was proposed by McCree and Barnwell [73]. This coder, called the Mixed Excitation Linear Predictor (MELP) is based on the concept of multi-band voicing decisions. The MELP coders addresses the short comings of the conventional LPC-10 [90] coders by employing a more realistic excitation signal. The excitation signal in the MELP coder is essentially a mixture of impulses and noise generated at in different frequency bands ($\sim$ 5 bands). At the decoder, the excitation thus generated is filtered by the LP synthesis filter to reconstruct the speech signal. The MELP encoder includes an auditory based approach to multiband voicing estimation for mixed impulse and noise excitation, aperiodic impulses to account for creaky and diplophonic sounds [**?**], and more accurate models for representing the shape of the glottal flow velocity source. A 2400 bps version of the MELP algorithm that uses a 5-band model and aperiodic pulses was adopted as the new U.S.military standard in

1996 [2] and the NATO standard for future band-limited voice coder [87] [3] in 2003. Since many of the low bit-rate coding efforts presented in this chapter are based on the MELP coder, a detailed description of the MELP coding standard is provided in Appendix 1.

2.1.2.2 Hybrid coders

While waveform coders achieve high quality speech reconstruction at bitates over 16 Kbps, model based coders use a model to descibe the speech signal and then compress the parameters of the model at bit-rates below 8 kbps. Hybrid coders blend the successes of the waveform coders in achieving high quality reconstruction with the low bit-rate encoding abilities of model based speech coders. Like model based speech coders, the hybrid coders employ a suitable model for the speech signal. However instead of modeling the excitation signal parametrically, hybrid coders either encode their waveforms directly or by a procedure called analysis–by–synthesis. The former is the same principle as waveform coders such as ADPCM. In the analysis–by–synthesis procedure, the excitation waveform is chosen such that the output waveform of the model matches that of the speech signal.

The analysis–by–synthesis hybrid coding concept forms the basis of several commercially popular speech coders including the code-excited linear prediction vocoders (CELP) [7] [48], the multipulse-excited linear prediction vocoders (MP-LP) [6], and the regular pulse-excited linear prediction (RPE-LPC) coders [56]. Several cellular communication standards employ CELP based speech coders. The GSM mobile communication system has standardized several CELP based coders including the full-rate codec in 1987, the enhanced full rate (EFR) coder in 1996 and the adaptive multirate coder in 1999. The CELP based codecs used in North American digital cellular standards include the IS 641 for TDMA and IS-127 for CDMA systems and the recent scalable mode vocoder (SMV) [4] for CDMA2000. Analysis–by–synthesis sinusoidal model based coders introduced by McAulay [72] and Quatieri and multi-band

excitation vocoders by Griffin and Lim [40].

2.1.3 Variable rate speech coding

Personal and mobile communication systems use digital multiple access techniques such as TDMA and CDMA. The use of multiple access techniques have evoked a keen research interest in variable rate speech coders, which allow a flexible allocation of bits to different segments of the speech signal. By dynamically allocating the bits to different regions of the speech signal based on the information content of that region, variable rate speech coders allow the communication system to use the available bandwidth more efficiently. For instance variable bit-rate speech coders may be employed to reduce the co-channel interference and thus increase the capacity of cellular systems [69]. Variable rate forward error correction techniques may be employed to achieve higher coding gains in regions where the speech coder bit-rates are low.

Yet another technology that has been gainfully employing variable rate speech coders is Voice Over IP (VoIP), which is the practice of using packet based networks instead of the standard public switched telephone network to send voice data. Variable rate speech coders enable a VoIP system to vary the rate of transmission based on dynamics of the speech signal. This as–need allocation of bandwidth imparts greater efficiency to the VoIP networks. Use of variable rate coding paradigms also enable VoIP systems to provide a wide range of quality of service (QoS) [9]. Further, by employing variable rate coders, a great deal of flexibility is available for efficient error control mechanisms.

Recently variable-rate multimode CELP coders have emerged that redistribute the available bits among various parameters adaptively. An example of such a coding paradigm is the IS-96 standard, QCELP, for speech coding over CDMA networks [17], that uses an energy based threshold to vary the bit allocation. Yet another class of variable rate speech codes are based on phonetic class segmentation. A CELP based

speech coder using phonetic classification was proposed by Wang and Gersho [96]. In [94], a variable rate CELP coder with objective evaluation measures for segmentation was presented. A variable rate CELP coding with segmentation of speech frames into voiced and unvoiced segments was proposed in [30] and a phonetic segmentation with a classification and coding strategy based on [96] was proposed by [76]. Hagen *et al* proposed a voicing specific LPC quantization scheme for variable rate speech coding in [42]. A survey of variable bit-rate systems based on the CELP coding paradigm can be found in [36].

A variable rate MELP coder in which the LP parameters were coded using either the current or the previously encoded speech signal was presented in [70]. In [34], a variable frame rate MELP coder that employs adaptive frame selection using dynamic programming was presented. In this thesis, a novel variable MELP coder that employs phonetic segmentation information will be presented in Chapter 5. In Chapter 6, we present a variable rate MELP coder design in which the vector quantization codebooks used in MELP parameter encoding are dynamically re-ordered to reduce the entropy of the transmitted symbols.

2.1.4 Vector quantization in speech coding

Speech coding often involves efficient encoding of the parameters of the speech model. Typically the parameter that requires the most bits is the LP coefficients. Transparent coding of LP coefficients requires that there should be no audible distortion in the reconstructed speech due to error in encoding the LP coefficients [74]. Often, LP coefficient encoding involves vector quantization of equivalent representations of LP coefficients such as Line Spectral Pairs (LSP), and Log Area Ratios (LAR) [80]. In this section, we describe vector quantizers used in speech coding and their design.

Consider a database, $\mathbf{V}$, of n-dimensional LSF vectors. Let ζ be a set of integers.

A LSF VQ encoder can be thought of as a mapping

$$\mathbf{Q} : \mathbf{x} \rightarrow i, \tag{11}$$

that maps the vector $\mathbf{x} \in \mathbf{V}$ to an integer $i \in \zeta$. Typically, i is selected to be the index of the codevector $\mathbf{C}_i$ in a codebook $\mathcal{C}$ that minimizes a predetermined distortion measure $D(\mathbf{x}; \mathbf{C}_i)$. If the codebook has N codevectors $\{\mathbf{C}_i, i = 0, ...N-1\}$, then $\zeta \stackrel{\triangle}{=} \{0, 1, ..., N-1\}$. The VQ decoder,

$$\widehat{\mathbf{Q}} : i \rightarrow \mathbf{C}_i, \tag{12}$$

maps the index i back to the codevector $\mathbf{C}_i$. Thus the VQ codec quantizes $\mathbf{x}$ to $\widehat{\mathbf{Q}}\big(\mathbf{Q}(\mathbf{x})\big) = \mathbf{C}_i$.

In the training mode, the codebook $\mathcal{C}$ is designed to minimize the expected value of D for all $\mathbf{x}$ belonging to a database of training vectors. The expected value of D is given by

$$\mathbf{D} = \sum_{i=0}^{N-1} \int_{\mathbf{V}_i} D(\mathbf{x}; \mathbf{C}_i) f_{\mathbf{x}}(\mathbf{x}) d\mathbf{x}, \tag{13}$$

where

$$\mathbf{V}_i = \{\mathbf{x} : D(\mathbf{x}; \mathbf{C}_i) \leq D(\mathbf{x}; \mathbf{C}_j), \text{ for all } j \in [0, N-1]\} \tag{14}$$

is the i^{th} partition of the database of training vectors. In many practical applications, the distortion measure $D(\mathbf{x}; \mathbf{C}_i)$ is chosen to be the square of the weighted Euclidian distance between $\mathbf{x}$ and $\mathbf{C}_i$, i.e,

$$D(\mathbf{x}; \mathbf{C}_i) = \|\mathbf{x} - \mathbf{C}_i\|^2 = \sum_{j=0}^{n-1} W_j(\mathbf{x}) \left(x_j - \mathbf{C}_{ij}\right)^2. \tag{15}$$

The weights, $W_j(\mathbf{x})$'s, must satisfy the condition, $W_j(\mathbf{x}) \geq 0$ and, in general, may be a function of $\mathbf{x}$ (thus the notation $W_j(\mathbf{x})$). These weights may be used to control the extent to which the each component of $\mathbf{x}$ impacts D. In other words, these weights may be used to control the coarseness of the vector quantizer along different

dimensions of $\mathbf{x}$. If the weights for a given $\mathbf{x}$ are represented by a diagonal matrix, such that

$$\mathbf{W}(\mathbf{x}) = \begin{bmatrix} W_0(\mathbf{x}) & 0 & \dots & 0 \\ 0 & W_1(\mathbf{x}) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & W_{n-1}(\mathbf{x}) \end{bmatrix}, \tag{16}$$

then

$$D(\mathbf{x}; \mathbf{C}_i) = (\mathbf{x} - \mathbf{C}_i)^T \mathbf{W}(\mathbf{x}) (\mathbf{x} - \mathbf{C}_i) \tag{17}$$

In the codebook design procedure, the optimal $\mathbf{C}_i$ is determined by setting the gradient of $\mathbf{D}$ with respect to $\mathbf{C}_i$ to 0.

$$\nabla_{\mathbf{C}_i} \mathbf{D} = 0. \tag{18}$$

Plugging in (14) and (17) in (18) we get,

$$\nabla_{\mathbf{C}_i} \mathbf{D} = -\int_{\mathbf{V}_i} 2\mathbf{W}(\mathbf{x}) (\mathbf{x} - \mathbf{C}_i) f_{\mathbf{x}}(\mathbf{x}) d\mathbf{x} \tag{19}$$

Rearranging the terms, we have

$$\mathbf{C}_i = \left[\int_{\mathbf{V}_i} \mathbf{W}(\mathbf{x}) f_{\mathbf{x}}(\mathbf{x}) d\mathbf{x}\right]^{-1} \left[\int_{\mathbf{V}_i} \mathbf{W}(\mathbf{x}) \mathbf{x} f_{\mathbf{x}}(\mathbf{x}) d\mathbf{x}\right] \tag{20}$$

Since the distribution $f_{\mathbf{x}}(\mathbf{x})$ cannot be determined analytically, a suitably designed and diverse training database is used to replace the actual distribution with an empirical one [35].

During the encoding process, a vector $\mathbf{x}$ is encoded according to the rule

$$\mathbf{Q}(\mathbf{x}) = i : D(\mathbf{x}, \mathbf{C}_i) \leq D(x, C_j) \text{ for all } j \in [0, N-1] \tag{21}$$

and the decoding process reconstructs $\mathbf{x}$ as $\mathbf{C}_i$. An optimal vector quantizer operates using a single large codebook with no constraints imposed on its structure.

The vector dimensions and codebook sizes required to implement such a VQ codec for transparent (high quality) speech coding are very large. Typically, a vector of 8–12

LP parameters derived from an appropriately windowed segment of speech will have to be coded with at least 24 bits to maintain good perceptual quality of the reconstructed segment [86] [74]. Thus, an unconstrained optimal vector quantizer with 2^{24} prototype vectors in its codebook will be required to encode these LP parameters. This renders the encoding complexity and the storage requirements prohibitively large.

Several structurally constrained VQ techniques reduce the complexity of implementation for a small degradation in the reconstruction quality compared to the optimal VQ. Structural constraints on the VQ may be imposed by splitting the vector into smaller vectors (split VQ) or by implementing the VQ encoder and decoder in multiple stages (multi-stage VQ). In [74], implementation of a split VQ of LP parameters is discussed.

2.1.4.1 Multi-stage vector quantizers

In multi-stage vector quantization (MSVQ), a vector of LP parameters is encoded by multiple VQ stage encoders. Often, these encoders are arranged in a cascaded structure so that each stage encoder encodes the error between the original vector and the reconstruction generated by all preceding VQ stage encoders [49]. The suboptimality of such a MSVQ arises from (i) the use of multiple codebooks to generate the reconstruction, (ii) stage–by–stage (sequential) search procedure for encoding the vectors and (iii) the use of the traditional sequential design algorithm for generating the stage codebooks [57].

A multi-stage vector quantizer (MSVQ) is a structurally constrained vector quantizer in which $\mathbf{x}$ is encoded by K successive VQ encoders $(\mathbf{Q}_1, \mathbf{Q}_2, ..., \mathbf{Q}_K)$. Thus, $\mathbf{x}$ is mapped onto a set of indices $I = \{i_1, ..., i_K\}$ by the K stage encoders. Here, i_k is the mapping generated by the k^{th} stage encoder. The K successive MSVQ decoders map this set I to the set of codevectors, $\mathbf{C}_I = \{\mathbf{C}_{i_1}^{(1)}, \mathbf{C}_{i_2}^{(2)}, ...\mathbf{C}_{i_K}^{(K)}\}$. The notation $\mathbf{C}_{i_m}^{(m)}$ represents the i_m^{th} codevector of the m^{th} stage codebook. Since the MSVQ codec quantizes $\mathbf{x}$ to the sum of codevectors $\sum_{m=1}^{K} \mathbf{C}_{i_m}^{(m)}$, the distortion $D(\mathbf{x}; \mathbf{C}_I)$ suffered

by $\mathbf{x}$ can be written as

$$D(\mathbf{x}; \mathbf{C}_I) = \|\mathbf{x} - \sum_{m=1}^{K} \mathbf{C}_{i_m}^{(m)}\|^2. \tag{22}$$

The optimal design and encoding procedure for MSVQ is described in detail in [57]. For the codebook design, it can be shown that the codevector indexed by j_k of the k^{th} stage, $\mathbf{C}_{j_k}^{(k)}$ can be given by

$$\mathbf{C}_{j_k}^{(k)} = \left[\sum_{\mathcal{I}_{j_k}} \int_{\mathbf{V}_{\mathcal{I}_{j_k}}} \mathbf{W}(\mathbf{x}) f_{\mathbf{x}}(\mathbf{x}) d\mathbf{x}\right]^{-1} \left[\sum_{\mathcal{I}_{j_k}} \int_{\mathbf{V}_{\mathcal{I}_{j_k}}} \mathbf{W}(\mathbf{x}) \mathbf{x} f_{\mathbf{x}}(\mathbf{x}) d\mathbf{x}\right] \tag{23}$$

where $\mathcal{I}_{j_k}$ is the set of indices whose k^{th} element is j_k, i.e.,

$$\mathcal{I}_{j_k} = \{i_1, \ldots, j_k, \ldots i_K\} \tag{24}$$

and the partition of the space is given by:

$$\mathbf{V}_{\mathcal{I}_{j_k}} = \left\{\mathbf{x} : D(\mathbf{x}; \mathbf{C}_{\mathcal{I}_{j_k}}) \leq D(\mathbf{x}; \mathbf{C}_{\mathcal{L}}) \quad \forall \mathcal{L}; \quad \mathcal{I}_l = \{l_1, \ldots l_K\}\right\}. \tag{25}$$

In the encoding mode, the optimal k^{th} stage index, j_k has to satisfy

$$D(\mathbf{x}; \mathbf{C}_{\mathcal{I}_{j_k}}) \leq D(\mathbf{x}; \mathbf{C}_{\mathcal{L}}) \quad \forall \mathcal{L}; \quad \mathcal{I}_l = \{l_1, \ldots l_K\} \tag{26}$$

Determination of the partition during the training (25) and the optimal index of the k^{th} stage (26) may be done by by an exhaustive joint search of all $\mathcal{I}_l$. Although a joint full search, instead of the sequential search, would be optimal, its complexity would be similar to that of the unconstrained VQ. Depending on the available computing power and constraints of the system, the sequential search may be replaced by other improved, but still suboptimal search procedures such as M-candidate search [5]. In the M-candidate search procedure, M codevectors that give the overall lowest distorion in the first stage is first computed. The second and the subsequent stages are searched M times and finally the out of the M paths, the one giving the overall lowest distortion is selected.

The sequential design of the stage codebooks is suboptimal since, while designing a given stage, it assumes that the subsequent stages are populated by zero vectors. This can be remedied by jointly designing the stage codebooks of the MSVQ [11]. The MSVQ scheme for LP parameters with joint design of codebooks and a joint search of the codebooks thus designed during the encoding process is presented in [57].

2.1.4.2 Split vector quantizers

The use of split vector quantizers for speech coding was reported in [74]. In a split vector quantizer, a n dimensional vector is split into L sub-vectors of smaller dimensions. Then L independent VQ encoders encode these sub-vectors and correspondingly L indices are transmitted to the receiver. At the decoder, the reconstructions corresponding to the sub-vectors are generated by a simple codebook look-up operation and the reconstructed sub-vectors are concatenated to reconstruct the input vector.

2.2 Speech enhancement

Speech enhancement has been a challenge for many researchers for almost four decades. The problem involves improving the performance of speech communication systems in noisy environments. Most speech enhancement algorithms focus on enhancement of the speech signal degraded by statistically independent, additive noise. In this section, we provide a brief review of speech enhancement algorithms is provided with emphasis on model based systems.

Speech enhancement systems may be classified into two basic types. Subtractive type algorithms estimate the short-time spectral magnitude of the speech by subtracting a noise estimation from the noisy speech [10]. Many variations and modifications to this basic approach have been reported [64] [95]. In [24] a minimum mean square error short-time spectral amplitude estimator was proposed. The second class of speech enhancement method is based on speech modeling. A speech enhancement algorithm that improved the quality and intelligibility of autoregressive model based

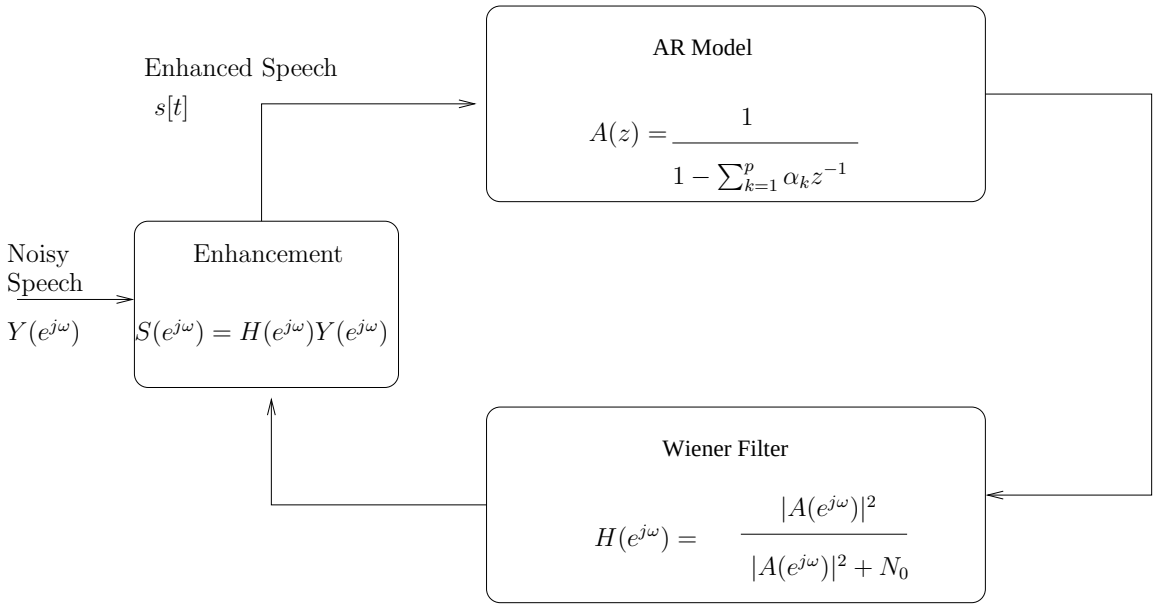

Figure 2. Autoregressive Model based Speech Enhancement

speech coders was first introduced by Lim and Oppenheim [63]. Lim and Oppenheim have suggested modeling the speech signal as a stochastic autoregressive (AR) process embedded in additive white Gaussian noise, and use this model for speech enhancement as shown in Figure 2. The algorithm is iterative in nature. It consists of estimating the speech AR parameters by solving the YuleWalker equations [44] using the current estimate of the speech signal, and then applying the (noncausal) Wiener filter to the observed signal to obtain a improved estimate of the desired speech signal. Hansen and Clements [43] proposed to incorporate auditory domain constrains in order to improve the convergence behavior of the Lim and Oppenheim algorithm.

The use of Kalman filtering for speech enhancement was first proposed by Paliwal and Basu [75], where experimental results revealed its distinct advantage over the Wiener filter, for the case where the estimated speech parameters were obtained from the clean speech signal (before being corrupted by the noise).

The basic structure of a Kalman filter is shown in Fig. 3. The Kalman filter addresses the problem of optimally estimating the state of a system that is governed

by the dynamic linear difference equation

$$\mathbf{x}[n] = \Phi[n]\mathbf{x}[n-1] + B\mathbf{u}[n] + \mathbf{w}[n] \tag{27}$$

with measurements

$$\mathbf{y}[n] = H\mathbf{x}[n] + \mathbf{v}[n], \tag{28}$$

where $\mathbf{w}[n]$ and $\mathbf{v}[n]$ are the measurement and model noises respectively. In a speech enhancement system employing a Kalman filter, the state of the system is formulated as a vector containing p samples of the speech signal and the state transition matric $\Phi[n]$ is made up of the AR model parameters. Gibson et al. [39] proposed to extend the use of the Kalman filter by incorporating a colored noise model in order to improve the enhancement performances for certain class of noise sources. Weinstein et al. [99] presented a time-domain formulation to model based speech enhancement problem. They represented the signal model using linear dynamic state equations, and then applyed the expectation maximization (EM) algorithm [20]. The resulting algorithm is similar in structure to the Lim and Oppenheim [63] algorithm, only that the noncausal Wiener filter is replaced by the Kalman filtering equations. In addition to that, sequential speech enhancement algorithms are presented in [99]. These sequential algorithms are characterized by a forward Kalman filter [51](Fig. 3) whose parameters are continuously updated. Lee et al. [58] extended the sequential single sensor algorithm of Weinstein et al. by replacing the white Gaussian excitation of the speech signal with a mixed Gaussian term that may account for the presence of an impulse train in the excitation sequence of voiced speech. Lee et al. examined the signal-to-noise ratio (SNR) improvement of the algorithm when applied to synthetic speech input.

Hidden Markov Models [81] are an useful class of models for the speech signal. HMM based speech enhancement was developed and studied in [21], [22] [25]. In [25], a maximum aposteriori (MAP) approach for enhancing speech signals which have

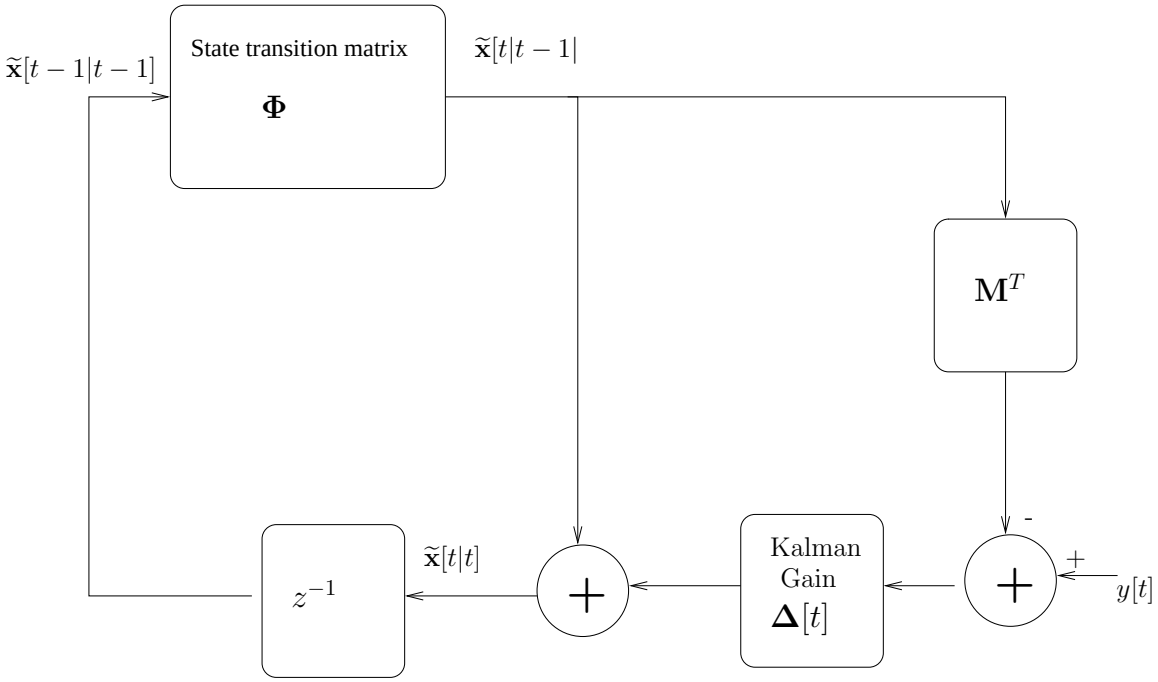

Figure 3. Kalman Filter for Recursive MMSE Signal Estimation

been degraded by statistically independent additive noise is proposed. The approach is based on statistical modelling of the clean speech signal and the noise process using long training sequences. For speech, HMMs with Gaussian AR output probabilities is used. The resultant algorithm is an EM [20] approach where a non-causal Wiener filter [44] is used to estimate clean speech and the expected value of the state of the system is evaluated using the HMM. An approximate MAP algorithm has also been suggested in which it is assumed that a unique sequence of states dominate the evaluation of the likelihood.

2.3 Joint source–channel coding

Several commercially popular speech coding standards have employed VQ [35] for encoding the parameters of the coders. One of the major problems associated with VQ is its sensitivity to errors in the received index due to noise in the communication channel. Although appropriate error correction codes may be employed to undo the

effects of such noise, it is often desirable to mitigate the effects without increasing the transmitted bit-rates. In this section, we provide a brief review of zero redundancy techniques that have been developed over the last two decades to design vector quantizers that are robust to channel noise.

Zero redundancy channel error protection techniques for VQ may be broadly classified into two distinct, though not mutually exclusive categories: Robust vector quantizers with optimally assigned codebook indices and channel-optimized vector quantizers. The former deals with the combinatorial optimization problem of appropriately assigning indices to the prototype vectors in the codebook so that frequently occurring transmission errors result in smaller reconstruction errors than transmission errors that are uncommon. In such techniques, the prototype vectors that constitute the codebook are first optimized for the signal source and then the problem of assignment of indices to these prototypes vectors is addressed [19]. Since the index assignment problem is "NP complete", several computationally less complex algorithms, such as simulated annealing [28], have been designed to come up with an appropriate permutation of the indices. Such algorithms result in either a globally or a locally optimal solution to the combinatorial optimization problem.

Channel-optimized vector quantizer [29] are designed by optimizing the VQ codebook for both the source and the communication channel. The design procedure is similar to the standard K-means [35] algorithm except that it incorporates the channel characteristics in the distortion measure. Thus, the partitions of the vector space and the prototype vectors in the codebook of the channel-optimized vector quantizers are designed to minimize the reconstruction error in the presence of channel noise.

Other error control techniques for VQ, include self organizing feature map based approach [62], channel-optimized predictive vector quantization [65], VQ by linear mapping of block codes [42], and algorithms that provide unequal error protection to the binary representation of the indices [31]. Recently, several adaptive channel

optimization techniques for vector quantized data have been reported. In this chapter an extensive survey of the state of the art speech coding, speech enhancement, and channel error protection algorithms is provided.

2.4 Evaluation of speech quality

The effectiveness of a speech coding system is determined in terms of its ability to preserve the information content and the natural quality of the speech signal. Similarly, the quality of the outputs of a speech enhancement system depends on its ability to suppress the interfering background noise without introducing any additional distortions or artifacts.

The evaluation of the quality of the speech signal is not a trivial task. It is well known that while two speech records may have very different waveforms, they may sound quite similar. For instance, model based speech coders aim to use models that capture the perceptual information in the speech signal rather than reproduce the speech waveform faithfully.

Several objective measures rate the quality of the reconstructed speech generated by a coding system or a noise suppression algorithm in terms of the fidelity of the reconstructed waveform or the similarity of short term power spectral density of the reconstruction and clean speech. Subjective measures evaluate the speech quality in terms of its naturalness, intelligibility, the background artifacts, and speaker identifiability [80]. A detailed description of these measures can be found in [86] [18] [79]. In this section, an overview of some of the measures that will be employed in this thesis is provided.

2.4.1 Objective measures

One of the most commonly used objective measures to evaluate the quality of a compression system or a noise removal system is the signal to noise ratio (SNR). If the signal samples are given by $s[t]$ for $t = 0, 1, 2, \ldots, T$ and the corresponding

samples of the signal whose quality is being assessed is $\widetilde{s}[t]$, then SNR is defined by

$$SNR = 10\log_{10}\frac{\sum_{t=0}^{T} s^2[t]}{\sum_{t=0}^{T}\left(s[t]-\widetilde{s}[t]\right)^2} \tag{29}$$

While the SNR is indicative of the long-term similarity in the waveforms of the two signals, it tends to ignore artifacts and errors in segments of the signal that are low in their energy level. This is accounted for in the segmental signal to noise ratio measure that averages the SNRs obtained from short duration segments of the speech signal. Thus if the segments are formed from τ samples of the signal, the SSNR is given by

$$SSNR = \frac{1}{L}\sum_{l=1}^{L} 10\log_{10}\frac{\sum_{t=0}^{\tau} s^2[l\tau+t]}{\sum_{t=0}^{\tau}\left(s[l\tau+t]-\widetilde{s}[l\tau+t]\right)^2} \tag{30}$$

Since the averaging in (30) is done after the log operation, the SSNR penalizes low SNR segments more severely [86].

The effects of quantization on the LP model parameters can be measured in terms of the log spectral distortion. If $A(z)$ is the unquantized LP analysis filter and $A(z)$ is its quantized version, then the log spectral distortion (SD) is given by,

$$SD \text{ (in dB)} = \int_{\omega_1}^{\omega_2} 20\log_{10}\left[\frac{|A(e^{-j\omega})|}{|\widehat{A}(e^{-j\omega})|}\right]d\omega, \tag{31}$$

where ω_1 and ω_2 correspond to 125 Hz and 3.1 KHz [57]. Another measure that may be used to evaluate the autoregressive models is the Itakura–Saito distance.

2.4.2 Subjective measures

The objective measures outlined above do not take into account the perceptual characteristics of the human ear. Therefore subjective quality evaluation using phonetically balanced speech records [33] are required to evaluate speech coders and enhancement systems. Subjective quality tests are usually based on opinions formed from comparative listening tests. The diagnostic rhyme test (DRT) is designed to measure

the intelligibility of the speech while diagnostic acceptability measure and the mean opinion score are used to evaluate the overall quality of the signal.

In this thesis, the subjective evaluation of the speech coding and enhancement algorithms is performed using a comparison category rating (CCR) test [1]. In these tests, K experienced listeners are asked to use headphones to listen to a series (say M) of pairs of utterances (Utterance A and B), and judge the relative quality of the second sample with respect to the first, on an integer scale of -3 to +3. The significance of this scale is listed in Table 1. The score for the k^{th} listener for the i^{th} pair is denoted $Q_{k,i}$.

CCR Scale	Interpretation
3	A is much better than B
2	A is better than B
1	A is slightly better than B
0	A and B sound the same
-1	B is slightly better than A
-2	B is better than A
-3	B is much better than A

Table 1. Rating scale for the comparison category rating (CCR) test where two speech records A and B are compared

For calibration purposes, each listener is presented M_{cal} pairs in which the utterance A and B are the same speech record. The number of scores, Q_{cal} that were marked 0 for these calibration pairs are recorded and each listener is assigned a weight $\omega_k = (\text{Number of } Q_{cal} = 0)/M_{cal}$. The average CCR score is then calculated as the weighted average

$$Q_{CCR} = \frac{1}{\sum_{k=1}^{K} \omega_k} \left[\sum_{k=1}^{K} \left(\frac{\omega_k}{M} \sum_{i=1}^{M} Q_{k,i} \right) \right] \tag{32}$$

The standard deviation of the scores is

$$\sigma_Q = \sqrt{\left(\frac{1}{\sum_{k=1}^{K} \omega_k} \right) \sum_{k=1}^{K} \frac{\omega_k}{M} \left[\sum_{i=1}^{M} \left(Q_{k,i} - Q_{CCR} \right)^2 \right]} \tag{33}$$

For a given finite number of listeners and utterances, to determine if Q_{CCR} is statistically significant, the single sample t-test described below is performed [84] [26].

Since only a finite number of listeners and utterances are used in conducting the CCR test, it is assumed that Q_{CCR} is an estimated value of the true weighted mean μ of the Student's–T distribution [84]. The following two hypotheses may be proposed for μ

- **Null hypothesis:** The null hypothesis is postulated as $\mu = 0$ and
- **Alternate hypothesis:** The alternate hypothesis is given by $\mu \neq 0$.

From (32) and (33), the t statistic t_{stat} is calculated as

$$t_{stat} = \frac{Q_{CCR}\sqrt{K}}{\sigma_Q} \tag{34}$$

The t-test rejects the null hypothesis *if and only if*

$$2\min(\mathcal{F}(t_{stat}; K-1), 1-\mathcal{F}(t_{stat}; K-1)) \leq \nu, \tag{35}$$

where $\mathcal{F}(a; b)$ is the cumulative distribution function of the Student's–T distribution with b degrees of freedom and ν is the significance level. The $(1-\nu)100\%$ confidence level in the Q_{CCR} score is given by

$$C_{(1-\nu)100\%} = Q_{CCR} \pm \frac{\sigma_Q}{\sqrt{K}}\acute{\mathcal{F}}(1-\frac{\nu}{2}, K-1), \tag{36}$$

where, $\acute{\mathcal{F}}$ is the inverse of the Student's-t cumulative distribution function.

CHAPTER 3

SPEECH ENHANCEMENT USING KALMAN FILTERS

The use of Kalman filters (KF) for estimation of clean speech from noisy measurements has been widely explored [75] [39] [32]. Typically, the KF formulation for speech signal estimation assumes that the speech signal can be modeled as a p^{th} order autoregressive (AR) process. To accommodate non-white spectral characteristics of the noise corrupting the speech, the noise signal is also modeled as a q^{th} order AR process. The state of the KF is usually defined to include p consecutive speech and q consecutive noise samples. The KF then provides a minimum mean square error (MMSE) estimate of the KF state at a time instance t, given the noisy measurement and the AR model for the time evolution of the state of the KF. The estimate of the clean speech signal can be derived from the estimated KF state.

The performance of such a KF system largely depends on the reliability of the estimates of the AR model parameters. Since the clean speech signal and the noise are unknown, standard procedures for AR model parameter estimation, such as the autocorrelation method, can not be employed. In this chapter, we develop and present a codebook constrained Kalman filtering system for estimation of speech in the presence of noise. In the proposed KF preprocessor, the AR model parameters for the clean speech and the noise signals are obtained from codebooks, $\mathcal{C}_s$ and $\mathcal{C}_v$, containing suitably designed prototype AR parameters of the speech and noise signals respectively. These codebooks are trained using the standard K-means clustering [35] of the AR parameters obtained from a database of clean speech and speech-free noise signals. During the operation of the KF, the appropriate AR parameters are selected from $\mathcal{C}_s$ and $\mathcal{C}_v$ every frame (10-40 msec duration) using an Expectation Maximization (EM) [20] algorithm. The mathematical formulation of the proposed KF preprocessor is presented in Section 3.1.

The codebook constrained Kalman filtering (CCKF) paradigm presented in this chapter can be employed as a stand-alone speech enhancement system or as a noise reduction pre-processor for a speech coding or a recognition system. A speech enhancement system using the CCKF is implemented and its performance is compared with that of the traditional unconstrained KF (UKF) and another popularly used speech enhancement system: the noise pre-processor (NPP) used with the standard MELP coder [2] [71]. This performance evaluation is done for various noise conditions and levels. Both objective (SSNR) and subjective (CCR) test results are provided in Section 3.2 to demonstrate the improved performance achievable when the CCKF is used for speech enhancement. Since the CCKF estimates gives a ML estimate of AR model parameters which are constrained to belong to a codebook derived from clean speech, the proposed system can be used effectively in conjunction with a model based speech coder. To evaluate the performance of the CCKF as a front-end to a speech recognition system, the Aurora2 noisy speech recognition tasks are performed. A brief description of the Aurora2 system for speech recognition using a simple back-end, and simulation results to show the improvement in recognition rates are provided in Section 3.3.

3.1 The codebook constrained Kalman filter

In this section, the mathematical formulation of the proposed speech signal estimator that uses a codebook constrained KF is presented. Let the noisy speech measurement at the time t be $y[t]$.

$$y[t] = s[t] + v[t] \tag{37}$$

The speech and the noise signals may be assumed to be statistically independent. Let the speech signal $s[t]$ and the noise signal $v[t]$ be modeled as Gaussian AR random

processes [23]. They may be expressed as

$$\begin{aligned} s[t] &= \sum_{j=1}^{p} \alpha_j s[t-j] + e[t], \\ v[t] &= \sum_{j=1}^{q} \beta_j v[t-j] + u[t] \end{aligned} \tag{38}$$

where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_p]$ are the p AR model parameters for the speech signal, and $\boldsymbol{\beta} = [\beta_1, \beta_2, ...\beta_q]$ are the q AR model parameters for the noise signal, $v[t]$. The signals $e[t]$ and $u[t]$ are independent Gaussian white noise signals with second order moments σ_e^2 and σ_u^2, respectively. Equation (38) can be written in vector-matrix notation as

$$\mathbf{x}[t] = \Phi \mathbf{x}[t-1] + G[t], \tag{39}$$

where

$$\begin{aligned} \mathbf{x}[t] &= \big[s[t-p+1], \ldots, s[t], v[t-q+1], \ldots, v[t]\big],^T \\ G[t] &= \big[0, \ldots, e[t], 0, \ldots, u[t]\big],^T \end{aligned} \tag{40}$$

and

$$\Phi = \begin{bmatrix} \Phi_s & \mathbf{0} \\ \mathbf{0} & \phi_v \end{bmatrix} \text{where}$$

$$\Phi_s = \begin{bmatrix} 0 & 1 & \ldots & 0 \\ 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_p & \alpha_{p-1} & \ldots & \alpha_1 \end{bmatrix} \text{ and } \Phi_v = \begin{bmatrix} 0 & 1 & \ldots & 0 \\ 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \beta_q & \beta_{q-1} & \ldots & \beta_1 \end{bmatrix}. \tag{41}$$

Let the autocorrelation matrix of $G[t]$ be $\boldsymbol{\Sigma} = E\left\{G[t]G[t]^T\right\}$. It should be noted that $\boldsymbol{\Sigma}$ contains elements from the set $\boldsymbol{\sigma} \doteq \{\sigma_e^2, \sigma_u^2\}$. The input, $y[t]$, is related to $\mathbf{x}[t]$ by

$$y[t] = \mathbf{M}\mathbf{x}[t], \tag{42}$$

where $\mathbf{M}$ is a $1 \times (p+q)$ vector with the 1 in the p^{th} position. The speech signal at time t can be derived from $\mathbf{x}[t]$ using

$$s[t] = \mathbf{M}\mathbf{x}[t] \tag{43}$$

3.1.1 The Kalman filter

If the AR model parameters, $\boldsymbol{\alpha}, \boldsymbol{\beta}$, and $\boldsymbol{\sigma}$ are known *a priori*, then a KF, whose state vector at t is $\mathbf{x}[t]$ and the state transition matrix is Φ, can be employed to estimate the clean speech signal. The AR model parameters can be derived if the clean speech signal and the residual noise signals are known. Since in a practical system these signals are unknown, an algorithm for the ML estimation of these AR parameters is described in Section 3.1.2. In this section, we provide the Kalman filtering equations for obtaining the sample-by-sample MMSE estimate of $s[t]$, assuming that the estimates of these AR parameters,

$$\widetilde{\Theta} = \{\widetilde{\boldsymbol{\alpha}}, \widetilde{\boldsymbol{\beta}}, \widetilde{\boldsymbol{\sigma}}\} \tag{44}$$

are available.

Let us assume that at a time instance $t-1$, the KF has completed the estimation of the state of the system, $\mathbf{x}[t-1]$, using all information available to it till time $\tau \leq t-1$. The information used in this estimate includes all measurements $y[t]$ for $t \leq \tau$. Let us denote this estimate of $\mathbf{x}[t-1]$ as $\widetilde{\mathbf{x}}[t-1|\tau]$. The estimate can be characterized in terms of the following co-variance matrix:

$$\mathbf{P}[t-1|\tau] \;=\; E\left\{(\mathbf{x}[t-1] - \widetilde{\mathbf{x}}[t-1|\tau])(\mathbf{x}[t-1] - \widetilde{\mathbf{x}}[t-1|\tau])^T\right\}. \tag{45}$$

If $\tau = t-1$, then $\widetilde{\mathbf{x}}[t-1|t-1]$ is the estimate of the state, $\mathbf{x}[t-1]$ given all necessary information till that time $(t-1)$.

Let $\widetilde{\Phi}_{\text{ML}}$ be the state transition matrix similar to (41), but constructed using the estimates of the AR model parameters $\widetilde{\boldsymbol{\alpha}}, \widetilde{\boldsymbol{\beta}}, \widetilde{\boldsymbol{\sigma}}$. It must be noted that the AR model parameters do not vary on a sample by sample basis, but only once every 10-30 msec.

Therefore, we can assume that then for a time-frame $T \equiv \{t = t_1, t_1 + 1, \ldots, t_2\}$, the $\widetilde{\Phi}$ holds good. Then the estimate of the state of the system, $\mathbf{x}[t]$, without including the measurement at t, $y[t]$ is given by

$$\widetilde{\mathbf{x}}[t|t-1] = \widetilde{\Phi}\widetilde{\mathbf{x}}[t-1|t-1] \tag{46}$$

The measurement of the noisy sample at t, $y[t]$, is given by (42). The set of Kalman filtering equations for the estimation of the state of the system at t, including the measurement $y[t]$ is given by

$$\widetilde{\mathbf{x}}[t|t] = \widetilde{\mathbf{x}}[t-1|t] + \Delta[t](\mathbf{Y}[t] - \mathbf{M}^T\widetilde{x}[t-1|t-1]) \tag{47}$$

$$\text{where} \quad \Delta[t] = \widetilde{\Phi}\mathbf{P}[t|t]\mathbf{M}\left[\mathbf{M}\mathbf{P}[t|t]\mathbf{M}^T\right]^{-1}, \tag{48}$$

$$\text{and} \quad \Lambda[t] = \widetilde{\Phi} - \Delta[t]\mathbf{M}^T. \tag{49}$$

$$\mathbf{P}[t+1|t+1] = \Lambda[t]\mathbf{P}[t|t]\widetilde{\Phi} + \widetilde{\mathbf{\Sigma}}. \tag{50}$$

$\Delta[t]$ is defined as the Kalman gain, and $\widetilde{\mathbf{\Sigma}}$ is the estimate of $\mathbf{\Sigma}$.

If the AR parameter estimates used in the Kalman filtering equations ((46)- (50)) are the ML estimates, then $\widetilde{s}[t]$ is the optimal MMSE estimate of the clean speech sample at t given by,

$$\widetilde{s}[t] = \mathbf{M}\widetilde{\mathbf{x}}[t|t]. \tag{51}$$

The EM algorithm for the ML estimation of the AR parameters is given in the following section.

3.1.2 Codebook-constrained ML estimation of AR parameters

The performance of the CCKF largely depends on the reliability of the estimates of the AR model parameters of the clean speech and the residual noise signals, but in a practical system, the true AR model parameters for use in the CCKF are unavailable. In this section, an iterative EM algorithm for obtaining the ML estimate of the AR model parameters from the noisy speech input to the CCKF for the time-frame

$t_1 \leq t \leq t_2$ is presented. It may be noted that while the KF operates on a sample-by-sample basis, the AR model parameters used by the CCKF may be updated on a frame-by-frame basis since these parameters tend to be stationary over short periods of time (10–40 msec).

Let us define the frame $\mathbf{Y} \doteq \{y[t], t_1 \leq t \leq t_2\}$, $\mathbf{s} \doteq \{s[t_1], s[t_1+1], \ldots, s[t_2-1], s[t_2]\}$, $\mathbf{V} \doteq \{v[t], t_1 \leq t \leq t_2\}$, and the set of AR parameters for this frame be denoted $\Theta = \{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\sigma}\}$. If $f(\mathbf{Y}; \Theta)$ is the PDF of $\mathbf{Y}$ parameterized on Θ, then the ML estimate of Θ is given by

$$\Theta_{ML} = \underset{\Theta}{argmax} \log[f(\mathbf{Y}; \Theta)]. \tag{52}$$

Defining the complete data log-likelihood function [20] as $\log[f(\mathbf{s}, \mathbf{V}; \Theta)]$, the i^{th} iteration of the EM algorithm can be described in the following two steps:

• ***The E step*** involves the evaluation of the cost function

$$Q(\Theta, \widetilde{\Theta}^{(i)}) = E\left[\log f(\mathbf{s}, \mathbf{V}; \Theta) | \widetilde{\Theta}^{(i)}, \mathbf{Y}\right]. \tag{53}$$

Since the PDF $f(\mathbf{s}, \mathbf{V}; \Theta)$ represents an AR Gaussian density, (53) can be expanded as

$$\log f(\mathbf{s}, \mathbf{V}; \Theta) = \log f(\mathbf{s}; \boldsymbol{\alpha}) + \log f(\mathbf{v}; \boldsymbol{\beta}) \tag{54}$$

Let us consider the term $f(\mathbf{s}; \boldsymbol{\alpha})$ in (54) From [91], it can be written in terms of the following marginal densities,

$$\begin{gathered} f(\mathbf{s}; \boldsymbol{\alpha}) = \prod_{t=t_1}^{t_2} f(s[t] | \xi_s[t-1]; \boldsymbol{\alpha}) \\ \text{where } \xi_s[t-1] = \{s[t-1], s[t-2], \ldots, s[t-p]\} \end{gathered} \tag{55}$$

Since speech is assumed to be AR gaussian,

$$f(s[t] | \xi_s[t-1]; \boldsymbol{\alpha}) = \frac{1}{2\pi\sigma_s^2} \exp\left[-\frac{\left(s[t] - \sum_{j=1}^{p} \alpha_j s[t-j]\right)^2}{2\sigma_a^2}\right] \tag{56}$$

Similarly for the noise signal v,

$$f(\mathbf{v};\boldsymbol{\beta}) = \prod_{t=t_1}^{t_2} f(v[t]|\xi_v[t-1];\boldsymbol{\beta})$$
$$\text{where } \xi_v[t-1] = \{v[t-1], v[t-2], \ldots, v[t-p]\} \tag{57}$$

Since $v[t]$ is also assumed to be AR gaussian,

$$f(v[t]|\xi_v[t-1];\boldsymbol{\beta}) = \frac{1}{2\pi\sigma_v^2} \exp\left[-\frac{\left(v[t] - \sum_{j=1}^{q} \beta_j v[t-j]\right)^2}{2\sigma_v^2}\right] \tag{58}$$

Substituting (54)–(58) in (53)

$$Q(\Theta, \widetilde{\Theta}^{(i)}) = -\frac{t_2 - t_1}{2} \log \frac{\sigma_s^2}{\sigma_v^2} - \sum_{t_1}^{t_2} \left[\frac{E\left\{(s[t] - \sum_{j=1}^{p} \alpha_j s[t-j])^2\right\}}{2\sigma_s^2} + \frac{E\left\{(v[t] - \sum_{j=1}^{q} \beta_j v[t-j])^2\right\}}{2\sigma_v^2}\right]. \tag{59}$$

The second order statistics in (59) are obtained as follows: From (45) and (47), since $E\{\mathbf{x}[t]\} = \widetilde{\mathbf{x}}[t|t]$, we have,

$$E\{\mathbf{x}[t]\mathbf{x}[t]^T\} = \mathbf{P}[t|t] + \left(\widetilde{\mathbf{x}}[t|t]\widetilde{\mathbf{x}}[t|t]^T\right) \tag{60}$$

Thus, the second order term $E\{s[t-i]s[t-j]\}$ in (59) is the $(p-i, p-j)^{\text{th}}$ element of $E\{\mathbf{x}[t]\mathbf{x}[t]^T\}$, and $E\{v[t-i]v[t-j]\}$ is the $(p+q-i, p+q-j)^{\text{th}}$ element of $E\{\mathbf{x}[t]\mathbf{x}[t]^T\}$.

The $\widetilde{\Phi}$ and $\widetilde{\Sigma}$ used by the KF (48) - (50) to evaluate (45) and (47) is constructed using $\widetilde{\Theta}^{(i)}$.

• ***The M step*** determines the set of AR parameters that maximizes the likelihood function

$$\widetilde{\Theta}^{(i)} = \underset{\Theta}{argmax}\ Q(\Theta, \widetilde{\Theta}^{(i)}). \tag{61}$$

The optimal AR parameters $\boldsymbol{\alpha}$ corresponding to the clean speech are constrained to belong to a suitably designed codebook $\mathcal{C}_s$. This codebook is designed by the standard K-means clustering of AR parameters derived from a database of clean speech

signals. The other AR parameters, $\boldsymbol{\beta}_k$'s and $\boldsymbol{\sigma}$, are estimated by an unconstrained maximization of the likelihood function [39]. Although $\boldsymbol{\alpha}$ can also be estimated as described in [39], we observed that the perceptual quality of the estimated clean speech is remarkably better when it is constrained to belong to the codebook $\mathcal{C}_s$. These results are presented in the following section.

3.2 Evaluation of CCKF

The performance of the CCKF is evaluated both in terms of objective and subjective quality of the enhanced speech. The objective quality of the enhanced speech is measured by the segmental signal to noise ratio (SSNR) described in Section 2.4. The proposed CCKF system is compared with the following two speech enhancement algorithms: the standard noise pre-processor (NPP) which is used in conjunction with the military and NATO standard MELP coder [2] [3] and the unconstrained KF (UCKF) described in [32]. The NPP algorithm enhances the speech input signal by estimating the *a-priori* signal-to-noise ratio in the input signal using an adaptive limiting algorithm, and modifying the input signal based on the estimates. Details of the NPP algorithm may be found in [71]. The UCKF [32] is similar to the proposed CCKF except that the AR model parameters are estimated from the second order statistics generated by the KF and are not constrained to belong to a codebook.

The implementation of the proposed CCKF for the objective and subjective evaluation is as follows: The clean speech signal, sampled at 8000 Hz, was modeled as a 10^{th}-order AR Gaussian process, and the noise signals, $v[t]$was modeled as 7^{th}-order AR Gaussian process. The AR model parameters were re-estimated every 128 samples. Approximately 100,000 AR parameter training vectors for the codebook $\mathcal{C}_s$ were obtained from the TIMIT training database, randomly selected from both male and female utterances representing all eight dialects. Extensive informal listening revealed that a small codebook size yielded poor quality speech due to the lack of sufficient

spectral resolution. It was also observed that a codebook, $\mathcal{C}_s$, with 2^{14} sets of AR parameters was adequate for obtaining acceptable quality of the enhanced speech. During the operation of the CCKF, the AR model parameters corresponding to the speech signal were determined through a brute-force search in $\mathcal{C}_s$ for the parameter vector that minimized the likelihood function. It may be noted that while a single codebook and a brute-force search was used in our implementation, other codebook structures and search procedures may also be employed. The parameters (frame size, model orders etc) in the implementation of the UCKF are chosen to be the same as the ones described above for the CCKF.

Twenty clean speech files were selected from the TIMIT [33] testing database that included speech files from both male and female speakers belonging to all 7 dialect regions. To these speech files, five different noise signals from the NOISEX-92 database [93] were added at 5 different signal to noise ration (SNR) levels ranging from -5 dB to 15 dB. The noise types selected include: buccaneer, M109, destroyer ops, white and babble. The averaged Welch periodogram [44] of the spectra of these noise types, obtained by averaging the modified periodogram from 512 sample Hamming windowed noise frames with a frame progression rate of 256 samples, are shown in Figure. 4. The noisy speech files generated as described above are then processed by the candidate speech enhancement systems. In the Tables 2–6, the SSNR measures corresponding to different noise types are compared. In the column (a) of all these tables, the SSNRs of the noisy signal are provided. These serve as a baseline for evaluating the performance of the speech enhancement systems. In columns (b) and (c) the SSNRs of the outputs of the NPP and the UCKF are provided.

Input SNR (in dB)	SSNR (in dB)			
	(a)	(b)	(c)	(d)
-5	-9.66	-0.71	-2.43	0.91
0	-6.23	1.49	0.45	2.38
5	-2.50	3.65	3.48	4.58
10	1.06	4.83	6.57	7.18
15	4.99	5.72	9.84	10.17

Table 2. Segmental signal to noise ratio of (a) the noisy speech signal, (b) the enhanced output of the NPP system, (b) the enhanced output of the UCKF system, and (d) the enhanced output of the CCKF system. The original speech is corrupted by buccaneer noise.

Input SNR (in dB)	SSNR (in dB)			
	(a)	(b)	(c)	(d)
-5	-9.31	0.60	-2.99	1.76
0	-6.10	2.63	0.19	3.91
5	-2.28	4.32	3.48	6.09
10	1.46	5.25	6.66	8.45
15	5.26	5.86	9.96	11.14

Table 3. Segmental signal to noise ratio of (a) the noisy speech signal, (b) the enhanced output of the NPP system, (b) the enhanced output of the UCKF system, and (d) the enhanced output of the CCKF system. The original speech is corrupted by M109 noise.

From the results presented in Tables 2–6, it is evident that the proposed CCKF algorithm outperforms the NPP consistently, except in the case of babble noise. It may be noted that babble noise is speech-like and, therefore, constraining the AR parameters to a codebook trained on clean speech does not improve the estimate of clean speech. The CCKF also outperforms the UCKF at low input SNR levels (-5,0 and 5 dB) and its performance is comparable to that of the UCKF at higher SNR levels. The improved performance of the CCKF at lower SNR levels may be attributed to the estimate of the AR model parameters from a codebook which has been trained on clean speech AR model parameters.

The spectrograms of the original clean speech signal, the signal with additive Bradley high noise at 0 dB SNR and the enhanced speech are shown in Fig. 5 (a),(b),

Input SNR (in dB)	SSNR (in dB)			
	(a)	(b)	(c)	(d)
-5	-9.74	-0.09	-0.35	1.00
0	-6.25	2.03	2.03	2.45
5	-2.47	3.98	4.56	4.50
10	1.12	4.94	7.29	7.07
15	5.03	5.74	10.39	10.05

Table 4. Segmental signal to noise ratio of (a) the noisy speech signal, (b) the enhanced output of the NPP system, (b) the enhanced output of the UCKF system, and (d) the enhanced output of the CCKF system. The original speech is corrupted by Destroyer Ops noise.

Input SNR (in dB)	SSNR (in dB)			
	(a)	(b)	(c)	(d)
-5	-8.82	-2.53	-4.76	-3.11
0	-5.72	0.58	-1.78	-0.66
5	-1.95	3.00	1.51	2.22
10	1.80	4.68	5.06	5.54
15	5.72	5.62	8.79	9.15

Table 5. Segmental signal to noise ratio of (a) the noisy speech signal, (b) the enhanced output of the NPP system, (b) the enhanced output of the UCKF system, and (d) the enhanced output of the CCKF system. The original speech is corrupted by babble noise.

and (c), respectively.

To compare the perceptual quality of the CCKF algorithm with that of the UCKF and the NPP, Comparison Category Rating (CCR) listening tests were conducted (refer Section 2.4.2). In these tests, 15 participants including native and non-native English speakers were asked to use headphones to listen to a series of pairs of utterances, and judge the relative quality of the second sample with respect to the first, on an integer scale of -3 to +3. The pairs of speech files presented were obtained from the processed outputs of the candidate speech enhancement systems, viz., CCKF, NPP and UCKF. In all, each listener compared the performance of the proposed CCKF with that of the NPP and the UCKF by listening to 8 pairs of speech files for each noise condition. Of these 8 pairs, 4 pairs consisted of the outputs of CCKF

Input SNR (in dB)	SSNR (in dB)			
	(a)	(b)	(c)	(d)
-5	-9.47	-0.86	-1.27	1.13
0	-6.22	1.51	1.42	2.81
5	-2.57	3.43	4.31	5.04
10	1.15	4.91	7.34	7.68
15	4.91	5.82	10.47	10.60

Table 6. Segmental signal to noise ratio of (a) the noisy speech signal, (b) the enhanced output of the NPP system, (b) the enhanced output of the UCKF system, and (d) the enhanced output of the CCKF system. The original speech is corrupted by white noise.

and UCKF algorithms and the other 4 consisted of the the outputs of CCKF and NPP algorithms. The set of speech files presented to each listener were randomly selected from a large pool of the outputs of the candidate speech enhancement system. Additionally, each listener was calibrated by presenting the same speech utterance in the pair and the scores of the listeners were weighted according to the fraction of such pairs that were given a score of 0. The order of each pair were randomized to prevent potential psychological biases. The CCR scores, Q_{CCR}, are presented in Table 7. The statistical significance of these scores for a significance level $\nu = 0.05$ and the 95% confidence levels are evaluated using the single sample t-test as described in Section 2.4.2 and are included in Table 7. The results presented verify the superior performance of the CCKF system as compared to the UCKF and the NPP systems for most cases.

Noise Condition	(a)			(b)		
	Q_{CCR}	Significant ?	$C_{(1-\nu)100\%}$	Q_{CCR}	Significant ?	$C_{(1-\nu)100\%}$
Buccaneer	1.35	yes	1.27-1.42	1.37	yes	1.30-1.43
Destroyer ops	0.58	yes	0.50-0.66	0.48	no	0.43-0.53
M109	0.13	no	0.06-0.21	0.75	yes	0.72-0.78
White	1.52	yes	1.44-1.59	1.67	yes	1.60-1.73
Babble	0.60	yes	0.57-0.63	1.32	yes	1.29-1.34

Table 7. Comparison category rating (CCR) measures of the proposed CCKF with respect to (a) the NPP system and (b) UCKF system

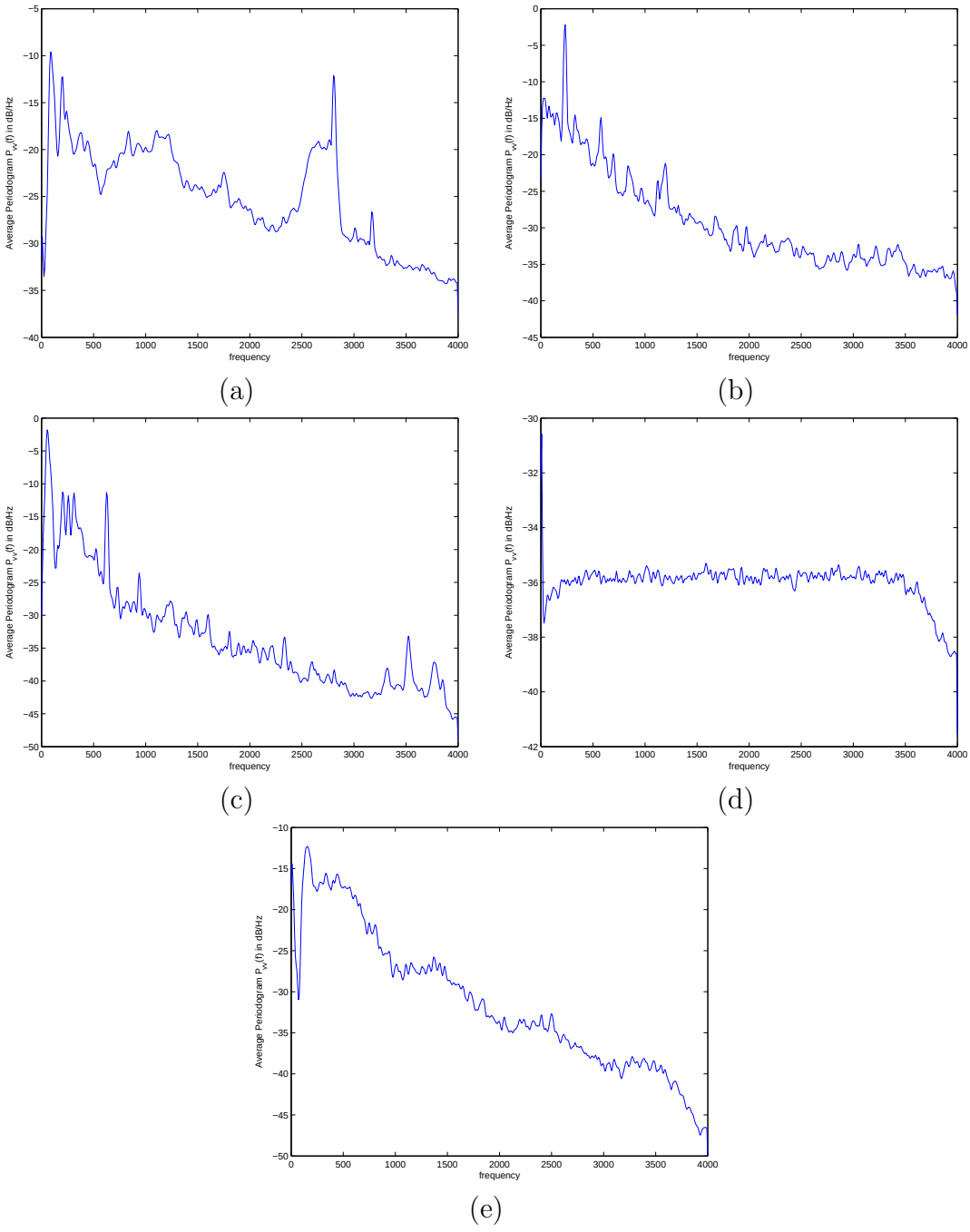

Figure 4. Averaged Welch periodogram of the noise types used in the evaluation of the CCKF: (a) Buccaneer, (b) Destroyer ops, (c)M109 Tank, (d) White and (e) Babble

3.3 Aurora-2 noisy speech recognition

Recently the design of automatic speech recognition (ASR) systems for use in personal and mobile electronic devices has been seeing a tremendous growth. The design of

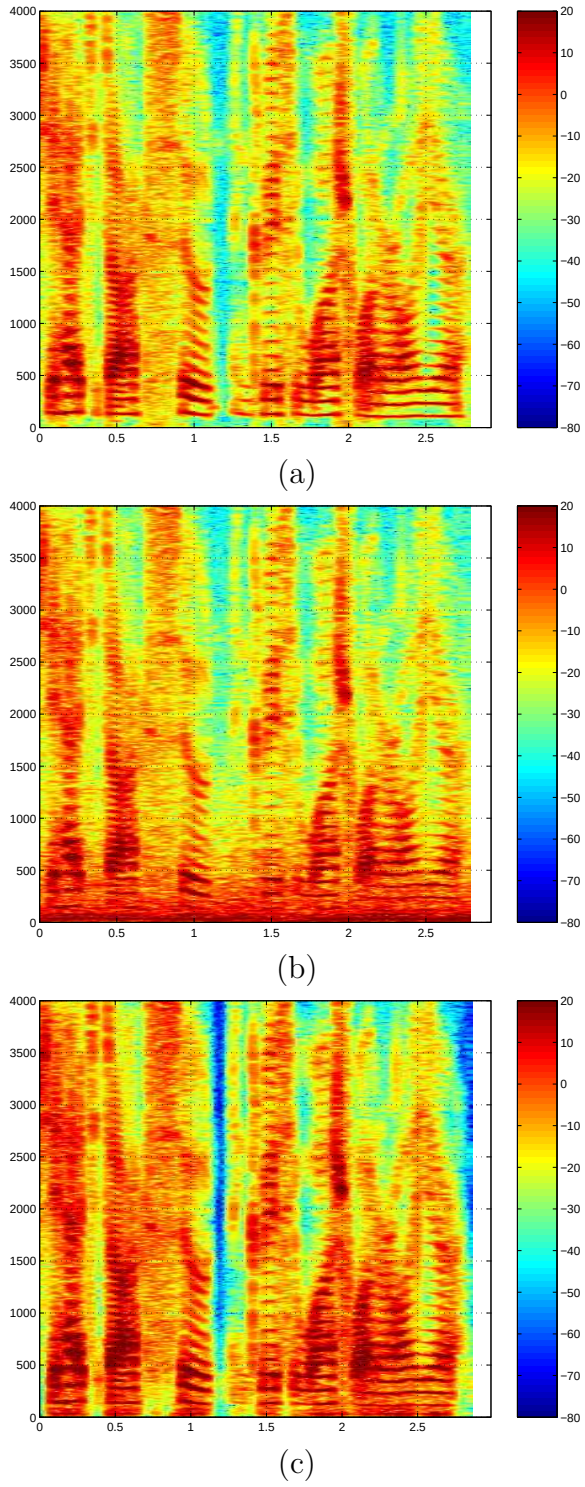

Figure 5. Spectrograms of (a)clean speech (b)speech with additive Bradley noise at 0 dB and (c)output of the CCKF

robust ASR systems for use in mobile environments poses several research challengers. First, these systems must perform without degradation in a variety of environmental conditions, where the input speech is corrupted by background noise. Second, the implementation of these systems is constrained by the limited resources available in wireless devices. In a distributed speech recognition (DSR) environment, features are extracted from the speech signal at the remote location and the recognition is performed in a centralized server.

One is the solutions to the problem of designing robust ASR systems is to employ noise suppression algorithms prior to the feature extraction by the DSR system. Alternatively, the recognition system can be trained so that the speech models match the noisy environment. While the former requires the incorporation of a suitable noise suppression algorithm in the front-end (feature extraction process), the later approach is related to the modification of back-end (the speech models that perform the recognition task).

3.3.1 Aurora-2 task

Given the need to have a common platform where researchers could test their noise pre-processing and ASR algorithms, and compare their results fairly, the Aurora DSR Working Group defined a set of connected digit string recognition experiments called the Aurora-2 task [77]. The basic Aurora-2 task consists of a standard front-end to extract the feature vectors and a standard back-end to perform the connected digit string speech recognition. Also a speech database was provided and an evaluation criterion was defined. This common platform is commonly referred as Aurora task. The Aurora-2 task provides a speech corpus referred as Aurora-2.0, which is a down-sampled subset of the TI Digits corpus. This database was artificially corrupted using different kinds of noise, including subway, babble, car, exhibition hall, restaurant, street, airport, and train station noise.

The Aurora-2 task defines two training modes: (a) *clean training mode* in which

the recognition engine is trained on clean data alone and (b) *multi-conditional training* where training is done using both clean and noisy data. The *clean* training database is composed of 8440 digit strings from TIDigits [61] that have been filtered by the G.712 characteristic filter without any addition of noise. The *multi-conditional* set consists of the same data as the *clean set*, but the data are divided into 20 subsets, each with 422 utterances. These 20 subsets represent 4 different noise conditions (suburban train, babble, car and exhibition hall) at 5 different SNR levels. The files are first filtered by the G.712 [46] filter prior to noise addition.

Three testing sets are provided for the evaluation of the Aurora-2 task. Each set has 4 subsets of 1001 utterances obtained from the TI Digits test database. The first testing set is *set A* that contains four sets of 1001 sentences, corrupted by subway, babble, car, and exhibition hall noises, respectively, at different SNR levels. Thus, the noise types included in this set are the same as those in the *multi-conditional* training. The second set, *set B* contains 4 sets of 1001 sentences each, corrupted by restaurant, street, airport, and train station noises at different SNR levels. These noise types are different from the ones used in the *multi-conditional* training. The test *set C* contains 2 sets of 1001 sentences, corrupted by subway, and street and airport noises. The data *set C* was filtered with the MIRS filter [46] before the addition of noise in order to evaluate the robustness of the algorithm under convolutional distortion mismatch.

3.3.2 Front-end noise suppression using CCKF

The performance of the CCKF described in Section 3.1 when incorporated in the front-end of the Aurora-2 task is presented in this section. In the experiments used in the performance evaluation, Aurora-2.0 speech database along with the ETSI Mel-cepstrum DSR (WI007) standard front-end version 2.0 were used. The standard front-end allows the extraction of a 39-dimensional feature vector composed of the 12 MFCCs (MFCC of order 0 is not included), logarithmic frame energy, and their first

and second order derivatives. The back-end consists in a whole word left-to-right continuous density hidden Markov model (CDHMM) where a single word is represented by 18 states, and each state has three diagonal covariance Gaussian mixtures. The search engine of HTK 3.0 toolkit was used to perform the experiments, and the default scripts provided in Aurora-2 CD-ROM were followed to set up the environment.

With the Aurora-2 task setup described above, two sets of experiments were performed. In the first experiment, the training was performed using features extracted from the *clean training* database. The testing speech files belonging to *set A*, *set B*, and *set C* were enhanced using the CCKF. Three iterations of the EM algorithm were performed and the speech and the noise processes were assumed be 10^{th} order AR processes.

In Fig. 6(a),(b), and (c), the accuracy of the recognition (defined as 100- word error rate) with and without the CCKF in the front-end are compared. In Table 8, the percentage improvements for different noise types and levels when the CCKF is used in the front-end over the baseline case where no enhancement is used in the front end are shown.

	Set A					Set B					Set C			
	Sub	Bab	Car	Exh	Ave	Res	Str	Apt	Sta	Ave	SubM	StrM	Ave	Ave
Clean	-3.74	0.00	0.00	0.00	-0.93	-3.74	0.00	0.00	0.00	-0.93	-10.47	0.00	-5.23	-1.79
20 dB	4.41	-8.32	-3.47	1.66	-1.43	-6.09	-6.34	-6.41	-5.11	-5.99	0.46	0.00	0.23	-2.92
15 dB	-7.07	70.39	38.96	-24.87	19.35	48.44	28.57	59.42	44.16	45.15	31.07	16.32	23.69	30.54
10 dB	47.18	66.94	69.35	31.92	53.85	38.29	50.29	56.11	59.2	50.97	36.02	24.79	30.41	48.01
5 dB	40.32	40.16	66.70	42.79	47.49	21.71	41.80	35.70	52.40	37.90	26.10	27.98	27.04	39.57
0 dB	21.95	14.79	32.96	28.24	24.48	8.86	18.28	11.64	20.48	14.81	13.62	15.45	14.54	18.63
-5 dB	3.21	-0.34	-1.58	4.13	1.36	-2.58	0.00	-4.54	0.00	-1.78	2.82	3.51	3.17	0.46
Ave	29.65	37.64	50.16	30.4	37.63	21.55	31.35	31.56	38.92	30.68	21.53	20.23	20.88	31.71

Table 8. Relative improvement in recognition accuracy with respect to baseline for the Aurora-2 task with clean training

In the second set of experiments, the recognition accuracy was evaluated when the back-end was trained using multi-conditional training data. In this case, the training data was also enhanced with the proposed CCKF. The testing speech files belonging

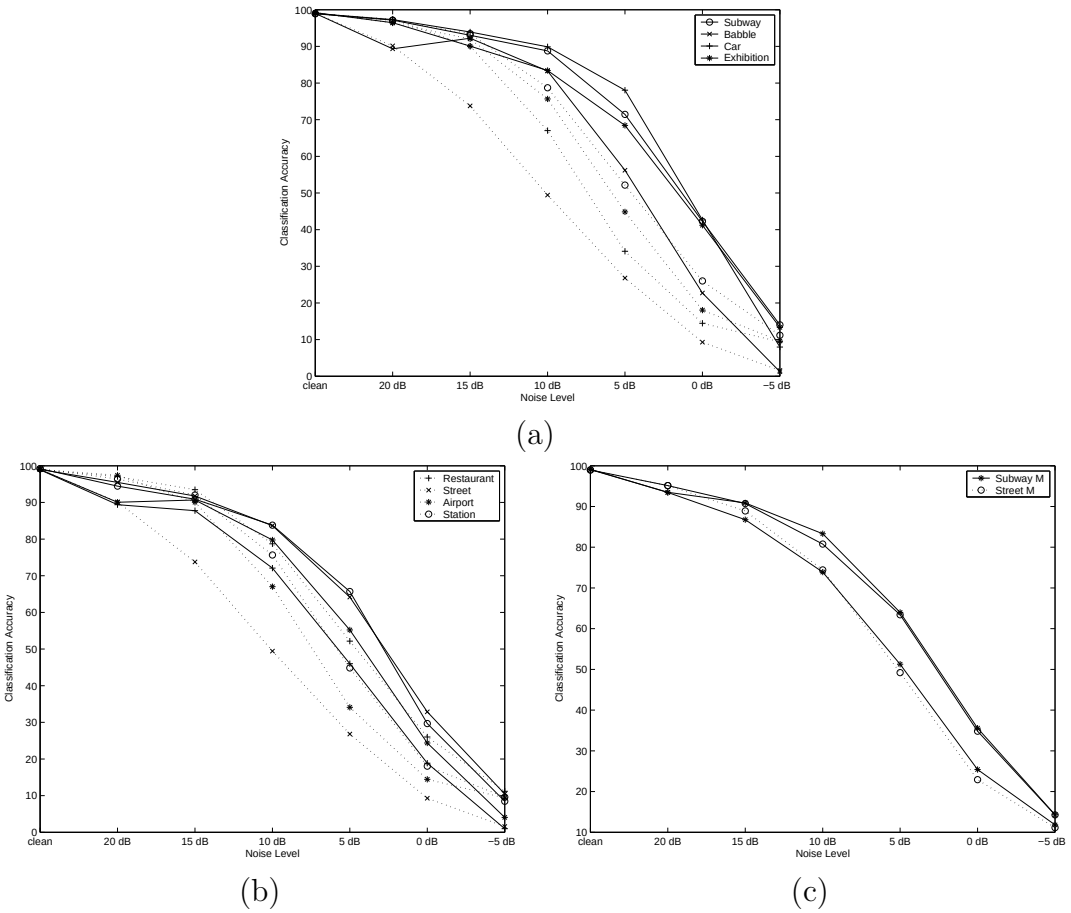

Figure 6. Recognition accuracy in the Aurora-2 task when the training is done using clean speech data and testing is done on CCKF enhanced files from (a) Set A, (b) Set B, and (c) Set C. Dotted lines refer to baseline case without CCKF in the front-end and solid lines refer to the case with CCKF in the front-end

to *set A*, *set B*, and *set C* were enhanced using the CCKF. Three iterations of the EM algorithm were performed and the speech and the noise processes were assumed be 10^{th} order AR processes.

In Fig. 7(a),(b), and (c), the accuracy of the recognition (defined as 100- word error rate) with and without the CCKF in the front-end are compared. In Table 9, the percentage improvements for different noise types and levels when the CCKF is used in the front-end over the baseline case where no enhancement is used in the front end are shown.

The recognition accuracy from the Aurora-2 experiments for *set A*, *set B*, and *set*

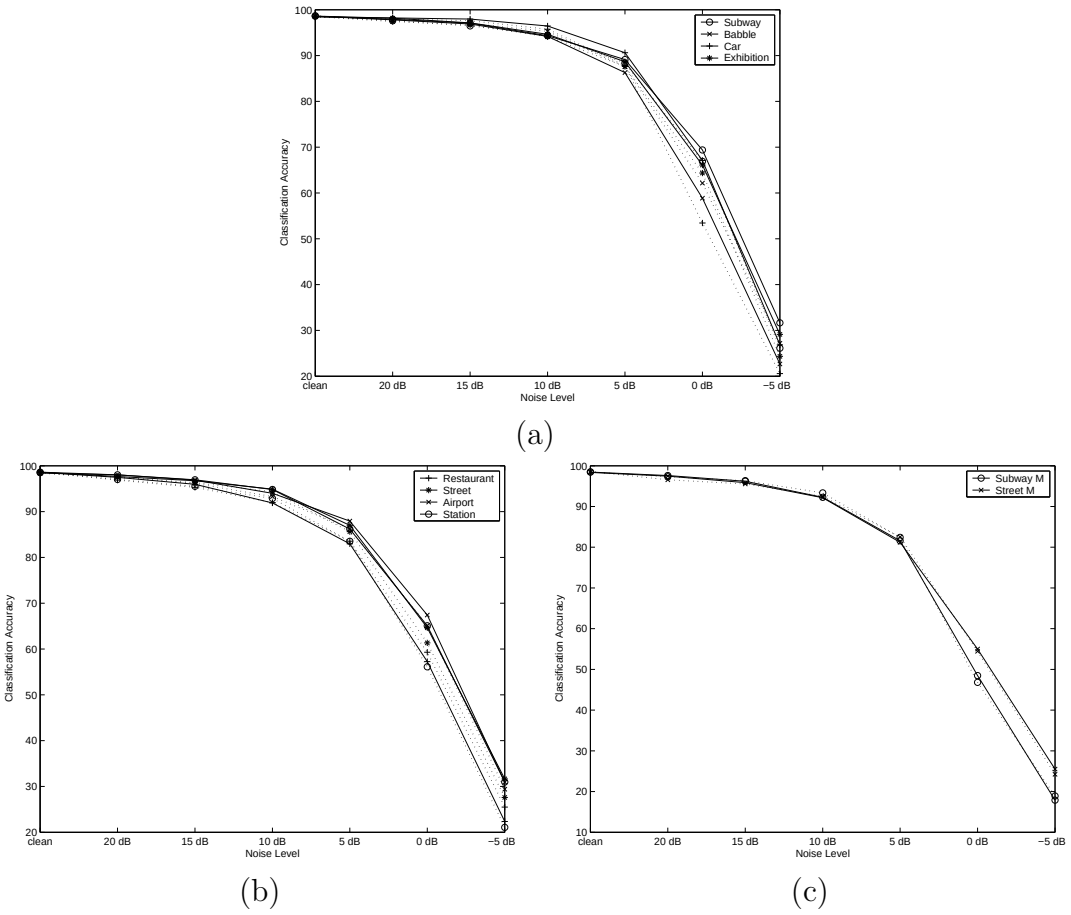

Figure 7. Recognition accuracy in the Aurora-2 task when the training is done using multi-conditional speech data enhanced using CCKF and testing is done on CCKF enhanced files from (a) Set A, (b) Set B, and (c) Set C. Dotted lines refer to baseline case without CCKF in the front-end and solid lines refer to the case with CCKF in the front-end

C under clean and multi-conditional training when the CCKF is used in the front-end as a pre-processor are summarized in Tables 10 and 11.

3.4 Summary

In this chapter, a Kalman filter based algorithm for enhancement of speech corrupted by additive background noise was provided. The proposed enhancement system assumes autoregressive models for the representation of the speech and the noise signal.

	Set A					Set B					Set C			
	Sub	Bab	Car	Exh	Ave	Res	Str	Apt	Sta	Ave	SubM	StrM	Ave	Ave
Clean	-6.82	4.05	1.86	7.95	1.76	-6.82	4.05	1.86	7.95	1.76	-2.67	-14.79	-8.73	-0.34
20	12.55	2.64	9.14	28.57	13.23	19.49	3.72	23.05	34.11	20.09	11.48	26.38	18.93	17.11
15	17.28	-5.07	15.06	13.81	10.27	13.83	11.65	19.33	31.77	19.15	-4.38	7.38	1.5	12.07
10	-2.16	-23.52	16.67	8.83	-0.05	-0.87	9.56	10.28	27.35	11.58	-16.22	-4.8	-10.51	2.51
5	6.62	-10.2	23.03	8.47	6.98	-3.52	9.66	12.15	16.14	8.61	-4.72	-7.56	-6.14	5.01
0	7.52	-8.69	29.34	4.85	8.26	-4.99	8.51	6.65	20.4	7.64	3.12	1.21	2.16	6.79
-5	7.47	-6.19	7.88	6.33	3.87	-4.24	5.59	2.2	12.56	4.03	-1.25	1.68	0.22	3.2
Ave	7.2	-9.56	26.3	7.52	8.42	-1.94	8.86	9.74	21.35	9.59	-0.12	0.14	0.01	6.89

Table 9. Relative improvement in recognition accuracy with respect to baseline for the Aurora-2 task with multi-conditional training

Training mode	SetA	SetB	SetC	Overall
Multiconditional	88.84	87.59	83.78	87.33
Clean	75.73	69.68	73.13	72.79
overall	82.28	78.63	78.45	80.06

Table 10. Absolute performance in recognition accuracy with respect to baseline for the Aurora-2 task with multi-conditional training

Training mode	SetA	SetB	SetC	Overall
Multiconditional	8.42	9.59	0.01	6.89
Clean	37.21	31.49	20.63	31.86
overall	22.81	20.54	10.32	19.37

Table 11. Relative improvement in recognition accuracy with respect to baseline for the Aurora-2 task with multi-conditional training

The model parameters are estimated on a frame–by–frame basis using an expectation–maximization approach. The key to the success of the proposed algorithm is the constraint on the autoregressive models corresponding to the speech signal to belong to a codebook trained on autoregressive parameters obtained from clean speech signal. The proposed codebook constrained KF was compared with a similar design that imposed no constraints on the autoregressive parameters, and another state of the art noise reduction system using objective and subjective evaluation measures and the superiority of the proposed approach was demonstrated. The CCKF was used as a pre-processor in the front end of the Aurora-2 noisy speech recognition task and improvement in recognition rate over baseline was reported.

CHAPTER 4

FRAMEWORK FOR FUSION OF OUTPUTS FROM SPEECH ENHANCEMENT SYSTEMS

The design of speech enhancement system has been a widely researched area during the last five decades. Typically, speech enhancement systems assume that the noise corrupting the speech signal is additive and uncorrelated with the latter, i.e., if $s[t]$ is the clean speech signal and $z[t]$ is the noisy observation at a sample time instance t, then $z[t] = s[t] + n[t]$ and $E\{s[t]n[t]\} = 0$, where $n[t]$ is the noise. Speech enhancement systems seek to estimate the clean speech signal $s[t]$ from $z[t]$ by minimizing the expected value of a suitably chosen distortion function. The outputs of speech enhancement systems often have residual noise and other artifacts, which are difficult to characterize analytically. However, on a sample-by-sample basis, the estimate $y[t]$ of the signal $s[t]$ generated by a speech enhancement system can be assumed to have a residual noise signal $v[t]$ and can be expressed as $y[t] = s[t] + v[t]$.

Based on the distortion function chosen and the strategy adopted to minimize the same, different speech enhancement systems yield different estimates of the clean speech signal $s[t]$. Therefore, it would be desirable to develop a "data fusion" framework for optimally combining the outputs of different speech enhancement systems to obtain an improved estimate of the clean speech signal. The ability of a Kalman filter to obtain a minimum mean-square error estimate (MMSE) of a signal on a sample-by-sample basis, using one or more noisy observations, makes it ideally suited for such a framework.

In this chapter, a novel multiple-input Kalman filtering (MIKF) framework is presented that estimates the clean speech signal by fusion of outputs from other speech enhancement systems. The MIKF framework generates a sample-by-sample minimum mean-square error estimate of the clean speech signal from these outputs.

The mathematical foundation of the proposed framework is similar to the CCKF described in Chapter 3.

As with the CCKF, the proposed MIKF framework assumes that the clean speech signal is modeled as a Gaussian autoregressive (AR) processes. The residual noise in each input to the MIKF is modeled as an autoregressive (AR) process. The AR model parameters for the MIKF framework are estimated using an iterative Expectation-Maximization (EM) algorithm similar to the description in Section 3.1.2. The EM algorithm obtains a maximum-likelihood (ML) estimate of the AR model parameters. Again, the AR model parameters for the speech are constrained to belong to a codebook of suitably designed AR model prototypes, trained on a database of clean speech. Constraining the AR parameters via a codebook improves the quality and makes it easy to integrate the MIKF system with a speech coder.

In generating a sample-by-sample MMSE estimate of the clean speech, the MIKF automatically weights each of its inputs in inverse proportion to the amount of residual noise present in that input. However, it may be desirable to impose additional heuristic weights to each of the inputs, which can be determined externally to the MIKF framework based on measures such as perceptual quality or intelligibility. The proposed framework has the flexibility to allow such heuristic weighting in a time-varying manner. A detailed description of how the parameters of the MIKF can be chosen to implement this weighting is provided in Section 4.2. Furthermore, since the EM algorithm seeks to estimate optimally the AR parameters for the speech model and constrains them to belong to a codebook of prototype AR parameters, the MIKF framework is well suited to be efficiently used in conjunction with any model-based speech coder.

Section 4.3 presents the results of a simulation in which speech enhancement outputs from two independent speech enhancement systems and the original noisy signal are successfully fused using the MIKF framework to estimate the clean speech

signal. It is demonstrated that the estimate of the clean speech by the proposed system has a better segmental signal-to-noise ratio (SSNR) and perceptual quality than any of the inputs to the MIKF (which are the outputs of the speech enhancement systems).

4.1 Multiple-input Kalman filtering paradigm

The mathematical formulation of the MIKF framework, shown in Fig. 8, is similar to the CCKF described in Chapter 3. Therefore, in this chapter we present modifications to the mathematical framework of the CCKF that leads us to the MIKF paradigm.

At the sample time t, the MIKF takes the outputs $y_1[t], y_2[t], ..., y_K[t]$ from K independent speech enhancement systems or from other sources. Also at t, let the residual noise in the outputs $y_1[t], y_2[t], ..., y_K[t]$ be denoted $v_1[t], v_2[t], ..., v_K[t]$ respectively. In other words, on a sample-by-sample basis

$$y_k[t] = s[t] + v_k[t] \quad \text{for } k = 1, 2, \ldots, K. \tag{62}$$

Let $\mathbf{Y}[t] = [y_1[t], y_2[t], \ldots, y_K[t]]^T$ be a vector containing samples from the outputs of various speech enhancement algorithms and $\mathbf{V}[t] = [v_1[t], v_2[t], \ldots, v_K[t]]^T$.

4.1.1 AR models for speech and residual noise

As in the case of the CCKF, it is assumed that the speech signal $s[t]$ and the residual noise signals $v_k[t], k = 1, 2, \ldots, K$ can be modeled as Gaussian AR random processes. The AR model for the speech signal is given by (38). Similarly, for each residual noise signal $v_k[t]$, the AR model may be expressed as

$$v_k[t] \;=\; \sum_{j=1}^{q^{(k)}} \beta_j^{(k)} v_k[t-j] + u_k[t], \text{ for } k = 1, 2, \ldots, K \tag{63}$$

Note that $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_p]$ are the p AR model parameters for the speech signal, and $\boldsymbol{\beta^{(k)}} = [\beta_1^{(k)}, \beta_2^{(k)}, ...\beta_{q^{(k)}}^{(k)}]$ are the $q^{(k)}$ AR model parameters for the residual noise signal, v_k. The signals $e[t]$(refer (38)) and $u_1[t], u_2[t], \ldots, u_K[t]$ are independent

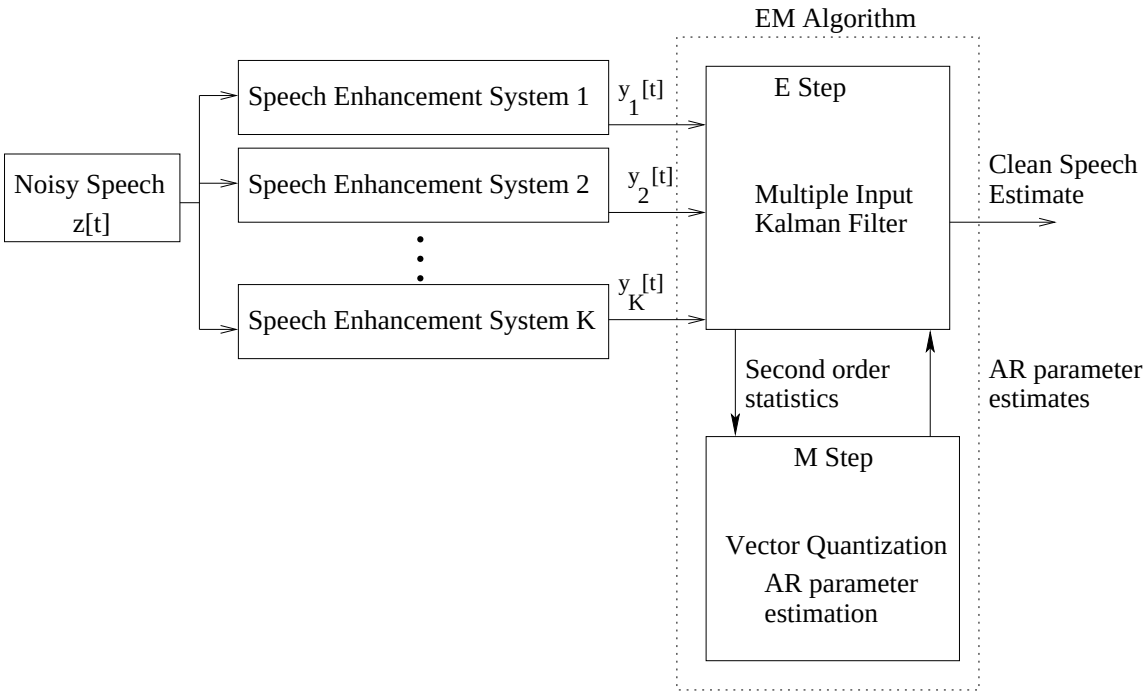

Figure 8. Multiple input Kalman filtering paradigm

Gaussian white noise signals with second order moments σ_e^2 and $\sigma_{u1}^2, \sigma_{u2}^2, \ldots, \sigma_{uk}^2$, respectively. Again, equation (63) can be written in vector-matrix notation as

$$\mathbf{x}[t] = \Phi\mathbf{x}[t-1] + G[t], \tag{64}$$

where

$$\mathbf{x}[t] = \big[s[t-p+1], \ldots, s[t], v_1[t-q^{(1)}+1], \ldots, v_1[t], \ldots, \tag{65}$$

$$v_K[t-q^{(K)}+1], \ldots, v_K[t]\big]^T \text{ and} \tag{66}$$

$$G[t] = [0, \ldots, e[t], 0, \ldots, u_1[t], \ldots 0, \ldots, u_K[t]]^T. \tag{67}$$

and

$$\Phi = \begin{bmatrix} \Phi_s & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \phi_{v1} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \phi_{vK} \end{bmatrix} \text{where}$$

$$\Phi_s = \begin{bmatrix} 0 & 1 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_p & \alpha_{p-1} & \dots & \alpha_1 \end{bmatrix} \text{ and } \Phi_{vk} = \begin{bmatrix} 0 & 1 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{q^k}^{(k)} & \beta_{q^k-1}^{(k)} & \dots & \beta_1^{(k)} \end{bmatrix}. \tag{68}$$

Let the autocorrelation matrix of $G[t]$ be $\mathbf{\Sigma} = E\left\{G[t]G[t]^T\right\}$. It should be noted that $\mathbf{\Sigma}$ contains elements from the set $\boldsymbol{\sigma} \doteq \{\sigma_e^2, \sigma_{u1}^2, \sigma_{u2}^2, \dots, \sigma_{uk}^2\}$. Also, since $\mathbf{\Sigma}$ is singular, the formulation presented herein can be implemented using reduced dimensions as described in [39]. Also in (64), The inputs to the MIKF framework, $\mathbf{Y}[t]$, are related to $\mathbf{x}[t]$ by

$$\mathbf{Y}[t] = \mathbf{M}\mathbf{x}[t], \tag{69}$$

where $\mathbf{M}$ is a $K \times (p + \sum_{k=1}^{K} q^{(k)})$ sparse matrix that implements (62). The speech signal at time t can be obtained from $\mathbf{x}[t]$ using

$$s[t] = \Psi\mathbf{x}[t] \tag{70}$$

where Ψ is a $(p + \sum_{k=1}^{K} q^{(k)}) \times 1$ vector, $\Psi = [0, 0, \dots, 1, 0, \dots 0]$, with the 1 in the p^{th} position.

4.1.2 Multiple-input Kalman filter

Since in a practical system these signals are unknown, an algorithm for the ML estimation of these AR parameters from the inputs to the MIKF is described in Section 4.1.3. If we assume that an estimate of these parameters is available, then a Kalman filter, whose equations are the similar to (46)–(50) to estimate the speech

signal from the outputs of the different speech enhancement systems. The variables in (46)–(50) for the MIKF system are determined as follows:

- The state of the system $\mathbf{x}[t]$ is given by (65).
- The state transition matrix is given by (68).
- The inputs to the MIKF are related to the state of the system via (69).

4.1.3 Codebook-constrained ML estimation of AR parameters of MIKF

In this section, the iterative EM algorithm for obtaining the ML estimate of the AR model parameters from the K inputs to the MIKF for the time-frame $t_1 \leq t \leq t_2$ is presented. It may be noted that while the Kalman filter operates on a sample-by-sample basis, the AR model parameters used by the MIKF may be updated on a frame-by-frame basis since these parameters tend to be stationary over short periods of time (10–40 msec). It may be noted that the This time frame is chosen such that the AR model parameters can be assumed to be approximately constant over this time frame.

Let us define the frame $\mathbf{Y} \doteq \{\mathbf{Y}[t], t_1 \leq t \leq t_2\}$, $\mathbf{V} \doteq \{\mathbf{V}[t], t_1 \leq t \leq t_2\}$, and the set of AR parameters for this frame be denoted $\Theta = \{\boldsymbol{\alpha}, \boldsymbol{\beta_1}, \boldsymbol{\beta_2}, \ldots \boldsymbol{\beta_K}, \boldsymbol{\sigma}\}$. If $f(\mathbf{Y}; \Theta)$ is the PDF of $\mathbf{Y}$ parameterized on Θ, then the ML estimate of Θ is given by

$$\Theta_{ML} = \underset{\Theta}{argmax} \log[f(\mathbf{Y}; \Theta)]. \tag{71}$$

Defining the complete data log-likelihood function [20] as $\log[f(\mathbf{s}, \mathbf{V}; \Theta)]$, the i^{th} iteration of the EM algorithm can be described in the following two steps:

• ***The E step*** involves the evaluation of the cost function

$$Q(\Theta, \widetilde{\Theta}^{(i)}) = E\left[\log f(\mathbf{s}, \mathbf{V}; \Theta) | \widetilde{\Theta}^{(i)}, \mathbf{Y}\right]. \tag{72}$$

Since the PDF $f(\mathbf{s}, \mathbf{V}; \Theta)$ represents an AR Gaussian density, and under the assumption that the speech and the residual noise signals are uncorrelated, the term

$\log f(\mathbf{s}, \mathbf{V}; \Theta)$ in (72) can be expanded as

$$\log f(\mathbf{s}, \mathbf{V}; \Theta) = \log f(\mathbf{s}; \boldsymbol{\alpha}) + \sum_{k=1}^{K} \log f(\mathbf{v_k}; \boldsymbol{\beta}_k) \tag{73}$$

Let us consider the term $f(\mathbf{s}; \boldsymbol{\alpha})$ in (73). As in the case of the CCKF, $f(\mathbf{s}; \boldsymbol{\alpha})$ can be written in terms of the marginal densities as in (56).

For each k, $f(\mathbf{v}_k; \boldsymbol{\beta}_k)$ can be written as follows in terms of the marginal densities:

$$\begin{aligned} f(\mathbf{v}_k; \boldsymbol{\beta}_k) &= \prod_{t=t_1}^{t_2} f(v_k(t) | \xi_{vk}(t-1); \boldsymbol{\beta}_k) \\ \xi_{vk}(t-1) &= \{v_k(t-1), v_k(t-2), \ldots, v_k(t-p)\} \end{aligned} \tag{74}$$

Since v_k is also assumed to be AR gaussian,

$$f(v_k(t) | \xi_{vk}(t-1); \boldsymbol{\beta}_k) = \frac{1}{2\pi\sigma_{v_k}^2} \exp\left[- \frac{\left(v_k(t) - \sum_{j=1}^{q^{(k)}} \beta_j^{(k)} v_k(t-j) \right)^2}{2\sigma_{v_k}^2} \right] \tag{75}$$

Substituting (73)–(75) in (72)

$$Q(\Theta, \widetilde{\Theta}^{(i)}) = -\frac{t_2 - t_1}{2} \log \frac{\sigma_s^2}{\prod_{k=1}^{K} \sigma_{vk}^2} - \sum_{t_1}^{t_2} \left[\frac{E\left\{ (s[t] - \sum_{j=1}^{p} \alpha_j s[t-j])^2 \right\}}{2\sigma_s^2} + \sum_{k=1}^{K} \frac{E\left\{ (v_k[t] - \sum_{j=1}^{q^{(k)}} \beta_j^{(k)} v_k[t-j])^2 \right\}}{2\sigma_{vk}^2} \right]. \tag{76}$$

The second order statistics in (76) are obtained from the (72) and (47) [39]. The $\widetilde{\Phi}$ and $\widetilde{\Sigma}$ used by the Kalman filter (48) - (50) to evaluate (72) and (47) are constructed using $\widetilde{\Theta}^{(i)}$.

• ***The M step*** determines the set of AR parameters that maximizes the likelihood function

$$\widetilde{\Theta}^{(i)} = \underset{\Theta}{argmax}\ Q(\Theta, \widetilde{\Theta}^{(i)}). \tag{77}$$

The optimal AR parameters $\boldsymbol{\alpha}$ corresponding to the clean speech are constrained to belong to a suitably designed codebook $\mathcal{C}_s$. This codebook is designed by the standard K-means clustering of AR parameters derived from a database of clean speech

signals. The other AR parameters, $\boldsymbol{\beta}_k$'s and $\boldsymbol{\sigma}$, are estimated by an unconstrained maximization of the likelihood function [39]. Again, as with the CCKF, although $\boldsymbol{\alpha}$ can also be estimated as described in [39], we observed that the perceptual quality of the estimated clean speech is remarkably better when it is constrained to belong to the codebook $\mathcal{C}_s$.

4.2 Heuristic weighting of inputs to the MIKF

Although the MIKF minimizes the mean-square error, it may be desirable to impose additional heuristic weights, based on measures such as perceptual quality or intelligibility, on each of the inputs. Thus, if it can be determined *a priori* that one of the inputs to the MIKF provides a perceptually inferior estimate, then a set of weights may be applied that suppresses the impact of that input on the clean speech estimate. The heuristic weights can be chosen so as to externally control the extent to which each $y_k[t]$ impacts the estimate of the clean speech signal. Further, the framework is flexible enough to enable the selection of these heuristic weights on a time-varying basis. The weighting of the inputs to the MIKF can be controlled by assuming that the signal $y_k(t)$ can be expressed as

$$y_k[t] = w_k s[t] + \mu_k v_k[t]. \tag{78}$$

The only modification required in the MIKF framework to incorporate weighting of the y_k's is a suitable modification of the matrix $\mathbf{M}$ according to (78).

In this section the influence of w_k and μ_k on the estimate of the clean speech is described. For the purposes of simplifying the analysis, but without loss of generality, let us assume that the MIKF estimates the clean speech from just two speech enhancement algorithms, i.e., $K = 2$. Selecting $p, q^{(1)}, q^{(2)} = 1$, the state vector $\mathbf{x}[t]$ becomes:

$$\mathbf{x}[t] = [s[t], v_1[t], v_2[t]]^T. \tag{79}$$

For the purposes of this simple analysis, we assume that $\mathbf{\Phi}$ and $P[t|t]$ are 3×3 diagonal matrices:

$$\Phi = \begin{bmatrix} \alpha_1 & 0 & 0 \\ 0 & \beta_1^{(1)} & 0 \\ 0 & 0 & \beta_1^{(2)} \end{bmatrix} \text{ and } P[t|t] = \begin{bmatrix} \pi_s & 0 & 0 \\ 0 & \pi_{v1} & 0 \\ 0 & 0 & \pi_{v2} \end{bmatrix}. \tag{80}$$

$\Delta(t)$ is a 3×2 matrix. In (47), the contribution to the state of the system $\widetilde{\mathbf{x}}[t|t]$ from the measured inputs is given by $\Delta(t)\mathbf{y}[t]$. Specifically, the state variable $s[t]$ is updated by the product of the first row of $\Delta(t)$ with the input vector $y[t]$ (47). From (48) - (50), the first row of $\Delta(t)$ is given by

$$\Delta_1(t) = \begin{bmatrix} \kappa_1 w_1 \pi_{v2} \mu_2^2 & \kappa_2 w_2 \pi_{v1} \mu_1^2 \end{bmatrix} \tag{81}$$

where κ_1 and κ_2 are constants, independent of the w_k's or the μ_k's. In (**??**), the term that relates $\mathbf{Y}[t]$ to the element $s[t]$ of the state $\mathbf{x}[t]$ is given by

$$\frac{1}{\kappa} \left\{ (\kappa_1 w_1 \pi_{v2} \mu_2^2) y_1[t] + (\kappa_2 w_2 \pi_{v1} \mu_1^2) y_2[t] \right\} \tag{82}$$

Therefore, the contributions of the y_k's to the estimate of clean speech $\widetilde{s}[t]$ may be controlled by varying the terms w_1, w_2, μ_1, and μ_2. These results can be easily generalized for K inputs.

4.3 Evaluation of the MIKF framework

To demonstrate the performance of the proposed system, a MIKF framework with three inputs is implemented. The three inputs are obtained from (a) the standard noise preprocessor used as a front end to the 2400 bps MELP coder (NPP) [2] [71], (b) an adaptive Wiener filtering (AWF) system [100], and (c) the original noisy speech. The rationale for choice (c) is that there may be some useful information in the noisy signal that is lost in the other two enhancement processes. Although only three inputs are used in the simulation results presented here, it should be emphasized that the

proposed system can estimate the clean speech signal from any number of waveform-based speech enhancement systems, provided they are approximately synchronized on a sample-by-sample basis.

To assess the performance of the proposed system, eight clean utterances were obtained from the TIMIT testing database [33], specifically one male and one female utterance from each of four North American English dialects, and downsampled to 8000 Hz. Samples of five different noise environments from the NOISEX-92 database [93] were similarly downsampled to 8000 Hz and added to each clean utterance to obtain SNRs varying from -5 to 20 dB.

The clean speech signal, sampled at 8000 Hz, was modeled as a 10^{th}-order AR Gaussian process, and the residual noise signals, $v_1[t]$, $v_2[t]$, and $v_3[t]$, were each modeled as 7^{th}-order AR Gaussian processes. The AR model parameters were re-estimated every 128 samples. Approximately 100,000 AR parameter training vectors for the codebook $\mathcal{C}_s$ were obtained from the TIMIT training database, randomly selected from both male and female utterances representing all eight dialects. Extensive informal listening revealed that a small codebook size yielded poor quality speech due to the lack of sufficient spectral resolution. It was also observed that a codebook, $\mathcal{C}_s$, with 2^{14} sets of AR parameters was adequate for obtaining acceptable quality of the enhanced speech. During the operation of the MIKF, the AR model parameters corresponding to the speech signal were determined through a brute-force search in $\mathcal{C}_s$ for the parameter vector that minimized the likelihood function as described in Section 4.1.3. It may be noted that while a single codebook and a brute-force search was used in our implementation, other codebook structures and search procedures may also be employed.

For purposes of the initial evaluation, each of the three inputs was weighted equally. If reliable phonetic segmentation or noise recognition is available, it may be possible to achieve greater performance by weighting the inputs differently, leveraging

knowledge of the enhancement methods' varying perceptual quality performance with respect to different phones or noise environments. Investigation of these weighting schemes will be presented in a future publication.

To quantify the performance of the MIKF system, the SSNRs (Section 2.4) of the enhanced and noisy speech signals were measured using the clean speech as the reference, and the means calculated for each SNR and noise condition. The differences in the SSNRs are tabulated in Tables 13–17, showing the SSNR improvement of the MIKF system over (a) the noisy speech, (b) the AWF output, and (c) the NPP output. Improvement is seen in all categories, and, as may be expected, the gains over each input improve both as the SNR decreases and as the stationarity of the noise increases. It is notable that the results verify the large SSNR gains that can be achieved by the MIKF, especially in adverse noise conditions (e.g., over 15 dB of gain in -5 dB M109 tank noise), but more significant is the fact that the MIKF achieves significant gains over both of the individual enhancement systems.

To assess the improvement in perceptual quality of the MIKF output over the inputs, Category Comparison Rating (CCR) listening tests were conducted. In these tests, experienced participants were asked to use headphones to listen to a series of pairs of utterances, and judge the relative quality of the second sample with respect to the first, on an integer scale of -3 to +3. Each pair consisted of the output of the MIKF and the corresponding output of either the AWF- or the NPP-enhanced inputs. The same set of 32 pairs of utterances were presented to each listener, but both the order of the 32 utterances and the order of each pair were randomized to prevent potential psychological biases. Two noise conditions were selected for testing, M109 and Buccaneer1, at 0 dB SNR. The Q_{CCR} indices were obtained by averaging the scores of all the listeners for each noise condition.

The results of the CCR test are presented in Tables 12, and they verify the significant improvement in quality over both the inputs, and in both noise conditions

Noise	AWF	NPP
Buccaneer 1	1.48	0.83
M109	1.25	0.50

Table 12. The Q_{CCR} index obtained when the output of the MIKF was compared to its inputs, (a) AWF and (b) NPP.

Input SNR	Δ SSNR improvement over		
	Noisy Speech	AWF system	NPP system
-5	13.7	4.2	3.8
0	12.7	4.6	3.1
5	11.2	4.7	2.5
10	9.8	4.2	1.9
15	7.3	3	1.4
20	5.3	1.5	1.0

Table 13. Improvement in segmental signal to noise ratio of the output of the MIKF over (a) the noisy speech signal, (b) the enhanced output of the AWF system, and (c) the enhanced output of the NPP system. The original speech is corrupted by white noise.

Input SNR	Δ SSNR improvement over		
	Noisy Speech	AWF system	NPP system
-5	15.5	5	4.9
0	13.8	3.9	4.5
5	10.9	3.9	2.7
10	10.2	4.6	2.2
15	7.1	3.8	1.2
20	5.5	2.4	1.2

Table 14. Improvement in segmental signal to noise ratio of the output of the MIKF over (a) the noisy speech signal, (b) the enhanced output of the AWF system, and (c) the enhanced output of the NPP system. The original speech is corrupted by M109 tank noise.

Input SNR	Δ SSNR improvement over		
	Noisy Speech	AWF system	NPP system
-5	13.4	4.3	3.9
0	11.9	3	2.8
5	10.4	3	1.9
10	8.3	3.7	2
15	6.8	3.7	1.3
20	5.1	2.4	1

Table 15. Improvement in segmental signal to noise ratio of the output of the MIKF over (a) the noisy speech signal, (b) the enhanced output of the AWF system, and (c) the enhanced output of the NPP system. The original speech is corrupted by destroyer ops noise.

Input SNR	Δ SSNR improvement over		
	Noisy Speech	AWF system	NPP system
-5	13.1	5.2	3.9
0	11.2	3.9	2.5
5	9.3	4	2
10	7.9	3.9	1.6
15	6.3	3.3	1.4
20	4.5	2.4	0.9

Table 16. Improvement in segmental signal to noise ratio of the output of the MIKF over (a) the noisy speech signal, (b) the enhanced output of the AWF system, and (c) the enhanced output of the NPP system. The original speech is corrupted by Buccaneer helicopter noise.

Input SNR	Δ SSNR improvement over		
	Noisy Speech	AWF system	NPP system
-5	9.7	2.8	3.1
0	8	2.2	1.9
5	7.6	3.1	2.1
10	6.5	3.4	1.4
15	4.6	2.8	0.9
20	3.5	2.1	0.7

Table 17. Improvement in segmental signal to noise ratio of the output of the MIKF over (a) the noisy speech signal, (b) the enhanced output of the AWF system, and (c) the enhanced output of the NPP system. The original speech is corrupted by babble noise.

tested. The improvement over the AWF system was more pronounced compared with the NPP. Furthermore, the quality of the MIKF output appears to show greater improvement in the less stationary Buccaneer1 noise.

4.4 Summary

This chapter has described and demonstrated a multiple-input Kalman filtering framework that fuses the outputs from multiple speech enhancement schemes to yield an improved estimate of the clean speech signal. The proposed MIKF paradigm is flexible, allowing any number of inputs, regardless of the noise sources, types, or levels, and also weighting of these inputs. Simulation results demonstrate the successful fusion of outputs from multiple speech enhancement systems in a wide range of SNRs and noise conditions, as measured in terms of objective and subjective criteria.

Many other considerations deserve more thorough investigation, for example, the choice of weights on each of the inputs to the MIKF, segmentation-based choice of weights, and the design of class-specific codebooks trained for different phonemes. Furthermore, work is in progress to integrate the MIKF framework with a speech coder and evaluate the subjective quality and intelligibility of the decoded speech.

CHAPTER 5

SPEECH CODING USING SEGMENTATION AND CLASSIFICATION

As discussed in Chapter 2, parametric coders such as MELP can encode the parameters of the LP model of the speech and the appropriate excitation source efficiently at bit-rates as low as 2400 bps. The 2400 bps MELP uses a frame size of 180 speech samples (sampled at 8000 Hz) and encodes the model and the excitation parameters using 54 bits per frame. One handicap of such coders is that, while they are designed to process speech, they do not use information about language extensively. Most coders distinguish only between voiced and unvoiced speech. Speech is generalized using an LPC model of the vocal tract and some combination of random and periodic excitation similar to that produced by the vocal system, but the characteristics of the language being spoken are seldom exploited.

To design coders that operate at lower bit-rates (below 2000 bps) or to improve the quality of present day low bit-rate speech coders, it is essential to explicitly take into account the language of the speech signal. Recently, very low bit-rate speech coders using the text–to–speech synthesis paradigm have been proposed [59]. These coders typically involve a recognition front-end that recognizes phonetic units of speech. The decoders are usually concatenative speech synthesis systems that reconstruct the speech signals by concatenation of the corresponding speech units followed prosody modification. Other speech coders employing ergodic HMM based synthesis have also been proposed recently [60]. Unfortunately, the performance of these coders largely depends on the efficiency of the speech–to–text and text–to–speech conversions and the accuracy of the synthesis models. Any error in the speech transcription causes catastrophic degradation in the quality of the reconstructed speech. Furthermore, these coders are largely experimental and their performance is tuned to a specific

user or to a small group of users [59].

In this chapter, speech segmentation and a broad phonemic classification information will be employed to improve the quality of low bit-rate speech coders and/or to enable lower bit-rate speech coding by efficient allocation of the bits. The proposed model will be developed within the MELP speech coding structure. Based on the assumption that speech segment classification information is available to us, we develop a wide gamut of techniques that modify the MELP parameter encoding process. Primarily, super-frame speech coding methods will be employed to reduce the redundancies in the representation of parameters of MELP speech coder. Also, enhancements to the current (2400 bps coders) using phonetic class specific information will be explored.

It is non-trivial to automatically generate a transcription of a speech signal which marks the phonemes spoken and their beginning and ending times. In this chapter, we propose using a more general process—to go from the speech signal to a higher level phonetic class transcription. This will allow us to combine similar classes of sounds at a level above the actual spoken content, which allows more detailed language modeling than most speech coders have employed. It is observed that, given a baseline coder and speech which has been segmented by phonetic class, a number of potential enhancements in the coder in terms of coding cost and quality of the reconstructed speech become possible. Using the TIMIT [33] database that include a phonetic level segmentation of the speech records and base coders drawn from the MELP family, proof-of-concept tests for several such enhancements are provided.

While the proposed coder is expected to enable good quality coding of speech at low bit-rates, its performance will be limited by the constraints of the application at hand. It is expected that some of the techniques presented in this chapter will involve buffering large segments of speech, which would result in delays in coding. For example, such coders will find extensive use in certain military applications where

very limited bandwidth is available for speech communication. Also, such coders may be employed to store large amounts of pre-recorded speech.

In the next section, the framework in which this initial testing was conducted, including the coder and acquisition of phonetic class segmentation will be presented. Section 5.2 will present several proposed speech coding improvements based on the availability of phonetic class segmentation and describe the results we saw in terms of the cost of coding and transmission and in terms of the quality of the reconstructed speech. Section 5.5 will summarize the results of this chapter and suggest future directions for development.

5.1 Framework

The availability of a phonetic class segmentation for a speech waveform enables us to make several enhancements to standard speech coders. The acquisition of phonetic class segmentation from speech is addressed in the next section. The base speech coders that is used to test these techniques are described in Section 5.1.2. Section 5.1.3 provides details about the test metrics employed here.

5.1.1 Phonetic class segmentation

In the techniques described in this chapter, only phonetic class segmentation information is required. In other words, only identification of both the phonetic class and its beginning and ending time in the waveform are required. It does not require lower-level phoneme segmentation (i.e., the actual spoken content.) The generalizations that can be made about the member phonemes (for example, /f/, /s/, /S/, and /T/) of a phonetic class (unvoiced fricatives) allow us to extract substantial savings from the parameter sets transmitted by a speech coder while avoiding a more difficult segmentation problem.

While automatic segmentation of speech into the desired classes [89] [102] can be

employed, for feasibility testing, phonetic class data extracted from the phoneme-level labels provided in the TIMIT database [33] have been used. These labels are compacted into ten phonetic classes, including: voiced and unvoiced fricatives, voiced and unvoiced plosives, affricates, vowels, nasals, liquids, glides, and silence. This information is provided to the coder in two ways depending on how it will be used. In the first case, a frame level decision about the phonetic class by selecting the dominant class across the frame is made. In the second, a sample level phonetic class determination aligned to the audio stream at 8 kHz is provided. TIMIT speech data is sampled at 16 kHz; it is resampled to 8 kHz for coding, except where otherwise noted.

5.1.2 Coders used for testing

These segmentation-based coding enhancements were implemented on the platform of two coders in the MELP family. The first of these is the standard MELP implementation at 2400 bps [73] [3]. Testing was also performed on an improved version of that coder, called MELP-I [26]. This variant focuses on accurate pitch detection and to pitch synchronous processing, using methods such as a circular LPC. The sampling rate, frame rate, and parameter encodings are the same as for MELP. Additionally, MELP-I has an object oriented framework well suited to rapid prototyping of speech coders that is advantageous for the type of work proposed in Section 5.2.

5.1.3 Testing

For initial testing, these phonetically modified speech coding enhancements were evaluated using three metrics. The first was the computational cost of the segmentation-based technique compared to that of the base coder (MELP or MELP-I) alone. The second was the cost of transmitting the parameters for the speech signal relative to MELP (savings compared to 2400 bps.) The third metric considered was the quality of the reconstructed audio from the enhanced coders as compared to that produced

by the base coder. For the purposes of this work, these were informal listening tests primarily targeted to detection of audible artifacts and obvious degradation of quality.

5.2 Super-frame coding of MELP parameters

The MELP coder is the US Department of Defense [2] standard algorithm for narrow-band secure voice coding applications. It has been found that the MELP coder operating at 2400 bps yields a significant speech reconstruction quality improvement over the CELP-10 standard [73].

For encoding speech at lower bit-rates using the MELP framework, super-framing techniques that exploit the redundancies in the MELP parameters may be used. A multiframe MELP coding approach developed was by Gersho et al that operates at 1200 bps [98] [97] and yields reconstructed speech at approximately the same subjective quality as the standard 2400 bps MELP.

5.2.1 Super-frames in 1200 bps MELP

The 1200 bps MELP encodes groups the MELP parameters of three consecutive frames of the standard 2400 bps MELP into a super-frame. The 1200 bps MELP quantization schemes are designed to efficiently exploit the super-frame structure by using Vector Quantization (VQ) and interpolation, taking into account the statistical properties of voiced and unvoiced speech. Each super-frame is categorized into one of several coding states according to the voiced/ unvoiced pattern of the super-frame.

5.2.2 Analysis of interframe redundancies

In this section, a super-framing approach similar to the one used in the 1200 bps MELP will be used for encoding the MELP parameters. However, instead of fixing the size of the super-frames to be three frames of the standard 2400 bps MELP, a flexible super-frame size will be used. The size of the super-frame will be determined based on the speech segmentation information available from the classification framework

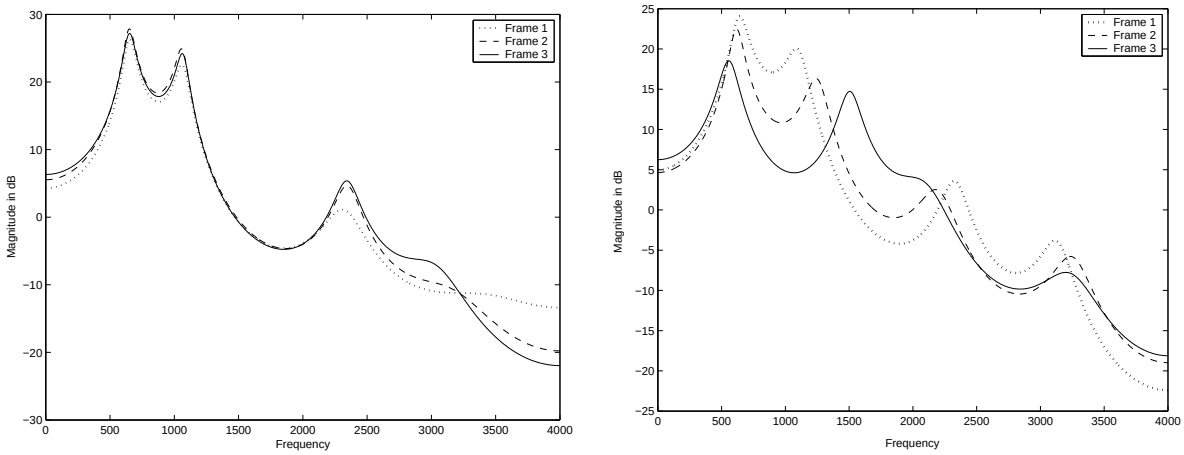

Figure 9. LPC spectra of (a) three consecutive frames belonging to the same phonetic class and (b) three consecutive frames used by the 1200 bps MELP coder

described in Chapter II. Thus the super-frames of the proposed coder will consist of consecutive frames belonging to the same phonetic class.

To analyze the redundancies in the MELP parameters of consecutive frames, the following experiment is performed. A speech file from the TIMIT database along with the phonetic segmentation of the speech is used to classify each 22.5 msec frames of speech into the following phonetic classes: *vowels, fricatives, stops, glides and semivowels, nasals, silent portions* and *transition*. The frames that were marked *transition* contained significant number of samples from two phonetic classes.

LPC Spectra In Figure 9 (a), the LPC spectra of three consecutive frames from the vowel /*IY*/ are plotted. For comparison purposes, the LPC spectra of the three consecutive frames in that would constitute a super-frame in the 1200 bps MELP coder are plotted in Figure 9 (b). The frames in the latter case are selected from the vicinity of the utterance of the same /*IY*/ as the former case. It is observed that the LPC spectra of the three frames belonging to the same phonetic class are similar to one another, while those belonging to the 1200 bps MELP super-frame exhibit a larger degree of variation.

Pitch Variance The MELP pitch parameter exhibits lower variance within a super-frame that is based on the phonetic segmentation than within the 1200 bps MELP coder. The average variance of the pitch within a super-frame for the proposed approach was found to be 39.6, while that for the 1200 bps MELP was found to be 71.8.

Fourier Magnitudes The MELP coding standard requires the transmission of 10 fourier magnitudes of the residuals every frame. To analyze the variance of the Fourier magnitudes within the super-frames of the proposed approach and the 1200 bps MELP coder, 10 speech files from the TIMIT database were selected. For the proposed coder, the super-frames were formed using the classification of the frames. For the 1200 bps MELP coder, the super-frames were formed with three consecutive MELP frames. For each super-frame, the variances of each of the 10 Fourier magnitudes are calculated. The comparison of the average variance of these Fourier magnitudes for the 1200 bps MELP and the proposed approach is shown in Figure 10. It may be observed that the variance of most of the Fourier coefficients is lower in case of the proposed super-framing approach as compared to the 1200 bps super-framing approach.

Gain The standard MELP coder requires two gain parameters to be transmitted every frame. The average variance of the two gain parameters within a super-frame for the 1200 bps MELP coder and the proposed approach is shown in Table 18. It is evident that the variance of the gain parameter is lower in the proposed super-frame approach than in the MELP 1200 super-frames.

	1200 bps MELP	Proposed approach
Gain 1	53.12	43.57
Gain 2	42.98	11.59

Table 18. Variance in the MELP *gain* parameter within a super-frame

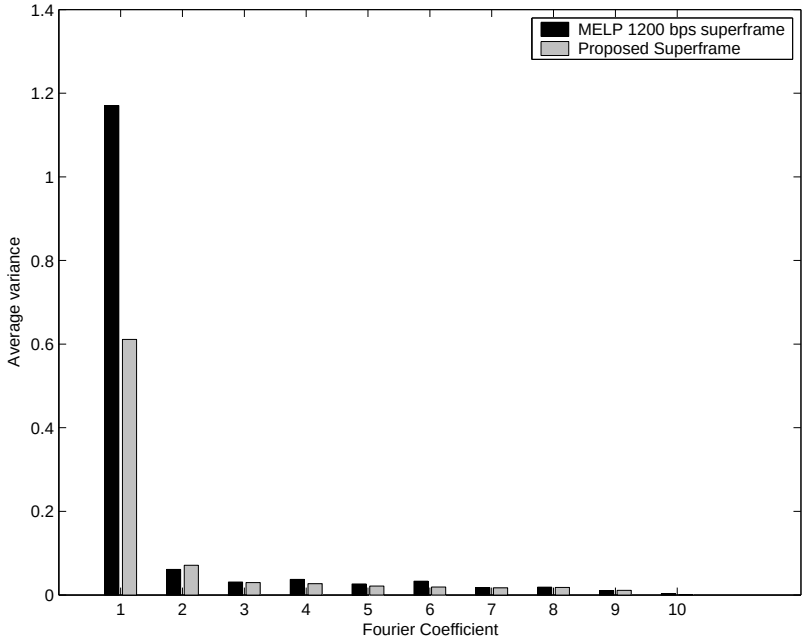

Figure 10. Average variance of Fourier magnitudes of the 1200 **bps MELP super-frames and the proposed segmentation based super-frames**

Other MELP parameters The MELP parameters, *band-pass voicing flag*, the *voiced/ unvoiced flag* and the *jitter flag* are basically binary flags. These parameters, within a super-frame may be efficiently encoded using loss-less coding algorithms such as run-length coding.

5.2.3 Super-frames based on classification

The redundancies described in the previous section are used to develop low bit-rate parameter encoding techniques using speech classification based super-framing. These technique were tested within the MELP framework in two ways. Standard MELP parameter sets were extracted from each 180 sample frame of speech. Then, based on the phonetic class segmentation, frames were grouped into phonetically similar super-frames composed of 1–3 MELP frames. Those super-frames were reduced to a single MELP parameter set, which was transmitted in place of the replaced frames. Alternatively, the super-frame size can be determined and then a single set of MELP parameters can be extracted from the frame as a whole (as a frame of from 180–540

samples.) At the decoder, the representative parameter set for the super-frame is simply duplicated to cover the required number of frames, and synthesis proceeds as usual.

This technique reduced the bit-rate of the transmitted parameters to as low as 900 bps. For all coders tested using this technique, the quality of the decoded speech was less than that of speech encoded with the 1200 bps version of MELPe. It is likely this could be adapted by improving the compaction of the parameters into the representative set. Instead of sending a single frame, we could add inter-frame information describing changes throughout the super-frame, much as MELPe does. It is additionally noted that some phonetic classes, such as unvoiced fricatives, responded better to this technique than did more rapidly evolving classes like diphthongs.

In terms of computation, selecting a single frame to transmit in place of a phonetic super-frame is transparent, although additional frame delay is introduced to allow three full frames to be considered (i.e., two additional frames of delay.) Calculation of MELP parameters from the larger, aggregated frames does increase the computational effort per super-frame, but the analysis of 1–3 smaller frames is eliminated at the same time.

5.3 Phonetic class-based codebooks

Speech frames with the same phonetic class exhibit considerable similarity in their MELP parameters, particularly when compared to the amount of similarity between the MELP parameters that represent a general frame of speech as described in the previous section. We can take advantage of this during speech coding when phonetic class segmentation is available by basing codebook selection on the phonetic class of the current frame of the speech signal.

The line spectral frequencies (LSFs) are the most expensive parameters to transmit in the MELP coder, requiring 25 bits per frame. Great savings in the number of

transmitted bits can be achieved by reducing the size of each LSF codebook by focusing the codebook on a specific phonetic class. At the same time, model of that class can be improved by focusing the codebook on one class only, rather than requiring it to be general enough to represent any frame of speech.

Two methods were used to create new LSF codebooks targeted to specific phonetic classes. The first was to train new vector quantization (VQ) codebooks based only on frames drawn from a single phonetic class. While more expensive, this method is well suited to rich and varied phonetic classes, such as vowels. New codebooks were generated from samples of vowel frames drawn from TIMIT. The resolution of those codebooks was selected to meet an average log spectral distortion (SD) close to 1 dB, fewer than 1% of the frames having more than 2 dB of SD, and no frames having more than 4 dB of SD. This allowed us to reduce the 25 bit LSF codebook used in MELP to as few as 14 bits for a codebook targeted only to vowels. The resultant average bit-rate was found to be 1925 bps. The CCR test (refer Section 2.4.2) was performed to compare the performance of the MELP coder with the replaced vowel VQ codebooks with the standard MELP coder and a score of 0.1889 was obtained. Further the t-test with significance level $\nu = 0.05$ confirmed that the Q_{CCR} was significant and the corresponding $C_{(1-\nu)100\%}$ was found to be 0.1819 to 0.1959. Therefore, the vowel codebook replacement not only resulted in reduction in average bit-rate but also improved the quality of the reconstructed speech.

The second codebook generation technique was based on the standard MELP multi-stage VQ (MSVQ) codebook. LSFs from frames in a single phonetic class were encoded using the MELP MSVQ, and the most frequently selected codewords from the first stage of the MELP MSVQ were used to build a smaller codebook for that class alone. This technique was used to build small codebooks from 16–128 words (4–7 bits), for smaller phonetic classes like unvoiced and voiced fricatives.

These techniques were tested using both the MELP and MELP-I coders. The

vowel codebook training resulted in little audible difference between the output of standard MELP and MELP with an altered vowel codebook. The MSVQ codebook reduction method used for the fricative classes in MELP-I did not audibly reduce the quality of the reconstructed speech at the decoder, even for codebooks as small as sixteen elements. The use of the reduced voiced and unvoiced fricative codebooks resulted in an estimated bit-rate of 2164 bps. Together, the estimated reduction in bit-rate for both codebooks combined was 1775 bps. The reduction in codebook size generally more than offset the need to transmit phonetic class information to the decoder to select the proper codebook.

There is little difference in computational cost between MELP with its original codebooks and using these alternate codebooks. While generating the codebooks is time-consuming and computationally expensive, it is a one time cost. During execution, the cost of searching those codebooks is generally lessened, since all of the tested codebooks were both smaller than the MELP MSVQ codebook and consisted of only a single stage.

5.4 Bandwidth extension for enhanced speech coding

Many phonetic classes, like vowels, exhibit most of their energy in the range from 0–4 kHz; this is easily captured at an 8 kHz sampling rate. Others, like fricatives, exhibit most of their energy above 4 kHz [27]; this information is lost when a coder like MELP is used to transmit speech. Bandwidth extension has been used to restore such lost information above 4 kHz [68], [41]. When phonetic segmentation is available, bandwidth extension can be targeted to those areas most affected by the limitations of the selected sampling rate without damaging other regions through unnecessary processing.

Fricative regions (both unvoiced and voiced) were identified in the coded audio stream using the phonetic class segmentation. Parameters in non-fricative regions

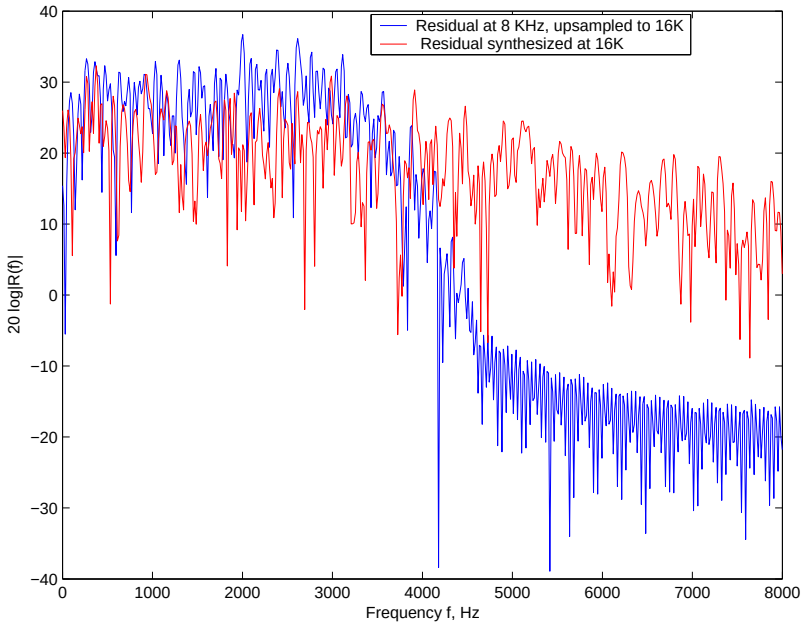

Figure 11. Bandwidth extension of reconstructed residuals of unvoiced fricatives

were extracted as usual for MELP at the standard 8 kHz sampling rate. When a fricative region was encountered, its parameters were extracted from the signal at 16 kHz (the original sampling rate in TIMIT.) The primary difference in extracted parameters is that, for a fricative, 20 LSFs were extracted from the higher rate signal. At the decoder, the residual in fricative regions was generated at 8 kHz, then upsampled to 16 kHz. It was then full wave rectified, mean subtracted and high pass filtered to extend the bandwidth of the residual. In Fig. 11, the power spectral densities (in dB) of the upsampled residual and the bandwidth extended residuals are shown. It may be noted that the full wave rectification followed by mean subtraction and high pass filtering boots the PSD in the 4-8 kHz band. Finally, the bandwidth extended residual was filtered by the larger set of 21 LPC coefficients. Segments of the speech with other phonetic classes did not undergo further processing; they were synthesized by MELP then upsampled to 16 kHz so that their sampling rate would match that of the bandwidth extended regions.

While not providing any real bit-rate reduction, this method does provide an

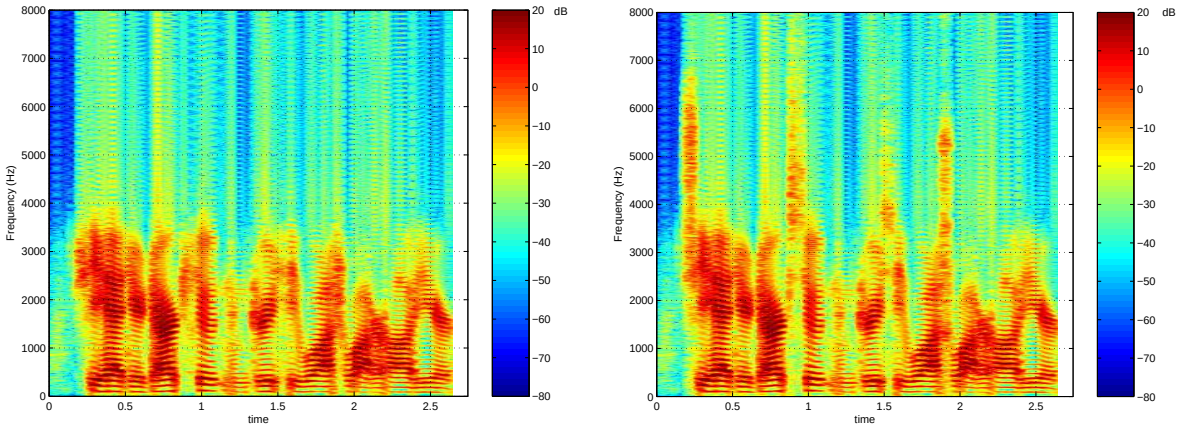

Figure 12. Spectrogram of (a) MELP-I output upsampled to 16 kHz and (b) bandwidth extension of unvoiced fricatives

improvement in the perceived quality of fricative regions by restoring the energy in the upper portion of the spectrum that was lost during coding. While the regions with extended bandwidth are sometimes noticeable in the context of longer speech segments, the novelty quickly fades, leaving only the perception of an improvement in those regions, and, therefore, in the speech overall. In Fig. 12(a) and (b) the wide-band spectrograms of the MELP coder upsampled to 16 kHz and that of the output of the bandwidth extension technique described above are compared. It may be observed that the later has significantly energy in the 4-8 kHz band during the unvoiced fricative regions. Computationally, the most expensive aspect of bandwidth extension is the need to operate at a 16 kHz sampling rate. Most additional processing comes from the larger fricative LSF set, creation of the bandwidth extended residual, and up-sampling of the remaining signal. It is unlikely that there is as much value in applying this technique to other classes, except perhaps the fricative portion of affricates, which is likely to suffer the same deficiencies.

5.5 Summary

In this chapter, it has been demonstrated that the combination of standard speech coding methods with phonetic class segmentation enables the implementation of a variety of enhancements to the chosen coder. With most of the techniques proposed in the previous section, it is observed that the modified MELP coder does not reduce the quality of the reconstructed audio when compared to MELP alone, while it does reduce the amount of data that must be transmitted by up to 50%. The most distorting method, simple phonetic class super-framing, shows a quality similar to 1200 bps MELPe (which has a comparable bit-rate.) Most of these methods are no more expensive in terms of computation than the base coder, though several introduce additional frame delay. None requires that more bits be transmitted than were required for the original MELP parameters and several do not require transmission of the phonetic segmentation.

Testing the system with an automatically generated phonetic class segmentation will also have to be done as that processing comes of age. This framework also provides fertile ground for further enhancement of a base speech coder using phonetic class segmentation. The obvious next step is to continue development of the discussed techniques. The phonetic class codebooks (Section 5.3) and super-frames (Section 5.2.3), while promising in their current form, could benefit significantly from additional development. Another goal is to combine the complementary methods into a single speech coder, so the benefits of each may be accrued. One possibility not addressed above is variable framing of the input based on its phonetic class; for example, transitions could be avoided by aligning the analysis frame boundaries with the transitions. We can also explore signal models specific to each class as was done for fricatives using bandwidth extension (Section 5.4) and demonstrated for plosives by Unno [92].

CHAPTER 6

DYNAMIC CODEBOOK RE-ORDERING FOR VQ OF CORRELATED SOURCES

In this chapter, a novel dynamic codebook re-ordering (DCR) procedure is presented that enables a very compact representation of the indices of the prototype vectors in a vector quantization (VQ) scheme for correlated source vectors. The DCR procedure causes the VQ indices corresponding to consecutive correlated source vectors to most likely belong to a small subset of all indices. Thus, the proposed DCR procedure dramatically reduces the entropy in the representation of the indices, which can be exploited for lossless compression of the VQ indices. The re-ordering of the codebook is done using a cost function that involves the previously selected prototype vectors from the codebook. Therefore the DCR procedure does not require the transmission of any additional information. Further, no additional distortion over the standard VQ is introduced by the incorporation of the DCR procedure in the VQ algorithm. Hence, a VQ system with DCR has the potential to achieve a significantly lower distortion at the same data rate as a standard VQ. The reduction in entropy is achieved at the cost of a moderate increase in the computational complexity associated with the re-ordering procedure. Simulation results using Gauss Markov source vectors are provided to demonstrate the entropy reduction achievable using DCR. To illustrate a practical application, the DCR procedure is applied to the VQ of the parameters of the MELP coder in Chapter 7. It will be shown that the DCR procedure may be employed to reduce the bit-rate of the MELP coder by nearly 40%, without introducing any additional distortions.

6.1 VQ symbol entropy

One of the fundamental findings of the theory of coding developed by Claude Shannon during the 1950s is that coding systems perform significantly better if they operate on sets of symbols or vectors rather than on individual symbols. Let $\mathbf{x}$ be an N dimensional vector of samples or some discrete features extracted from a signal. It may be recalled from Chapter 2 that in VQ, $\mathbf{x}$ is quantized to one of K pre-determined prototype vectors that are stored in a codebook $\mathcal{C}$. Let the codebook $\mathcal{C}$ contain the prototype vectors C_k, where the index $k \in \{0, 1, 2, \ldots, K-1\}$ indicates the location of C_k in $\mathcal{C}$. Typically, the codebook is stored in a memory and the physical address of the prototype vector C_k is its index k. The prototype vectors in the codebook are designed by the generalized Lloyd algorithm [35] that seeks to minimize the expected value of a suitably chosen distortion measure.

Let the source vector $\mathbf{x}$ be a random variable in a N dimensional vector space $\mathbf{V}$ and let ζ be a set of digital symbols. The VQ encoder can be defined as the mapping given by (11). In a traditional vector quantizer, the symbol i is selected to be the index k of the $\mathbf{C}_k$ in $\mathcal{C}$ that minimizes a distortion measure $d(\mathbf{x}; \mathbf{C}_k)$. Thus, in a traditional VQ system, if

$$k = \underset{j}{\operatorname{argmin}} \quad d\left(\mathbf{x}; C_j\right), \tag{83}$$

then $i \equiv k$. The VQ decoder reconstructs $\mathbf{x}$ using the prototype vector C_k. The VQ of $\mathbf{x}$ to C_k can be represented as a function Q,

$$Q(\mathbf{x}) = C_k. \tag{84}$$

and the output of the encoder for the input $\mathbf{x}$ is k.

Real life signals such as speech, image and video are non-stationary, or, at best, stationary over short periods of time or space. Therefore, to encode these signals, vectors are first derived from appropriately chosen short-duration frames of the signal. The sequence of vectors is then mapped to a sequence of symbols by the VQ process.

If we consider the source vectors derived from the signal as random variables, then we can associate a probability $p(i)$ with every $i \in \zeta$. Let p denote the probability mass function of the symbols, then the entropy of these symbols for a given signal source is

$$H = - \sum_{\text{all symbols } i} p[i]. \log_2 (p [i]) \tag{85}$$

In an optimally designed vector quantizer, if the statistics of the vectors used for training are similar to that of the vectors being encoded, the probability mass function (PMF) of the symbols in ζ corresponding to the source vectors tend to be flat [50], i.e.,

$$p[i] \approx \frac{1}{K}. \tag{86}$$

Consequently, the entropy in the distribution of these symbols is approximately $\log_2 K$.

It is well known that vectors derived from consecutive segments of most real world signals are strongly correlated. Hence, it can be expected that the quantized source vectors and the corresponding encoded symbols tend to be correlated as well. To reduce the entropy in the distribution of the VQ symbols by effectively exploiting this correlation between consecutive vectors, it is essential for the VQ codebook to have some kind of structure. In other words, the prototype vectors that are likely to get selected corresponding to consecutive source vectors must be arranged in the codebook such that the sequence of symbols generated can be effectively compressed by a lossless coding algorithm [83]. For instance, in a standard VQ scheme, reduction in the entropy of the symbols can be expected if the codebook is organized such that similar codevectors are assigned adjacent symbols. Such an optimal assignment of K symbols to K codevectors is an NP hard problem. While structured VQ systems such as tree structured VQ, entropy constrained VQ, etc., provide a scope for better lossless compression of the symbols than the standard VQ, it must be noted that for a given codebook size, K, and without any lossless coding of the indices, these VQ

schemes have a higher distortion than the standard VQ. The effectiveness of lossless compression will largely depend on the characteristics of the signal.

In [66], lossless compression of the standard VQ symbols derived from speech, image, and video signals was studied. Additionally, structurally constrained VQ such as memoryless tree structured VQ (TSVQ), pruned TSVQ, and entropy constrained VQ that are better suited for subsequent lossless coding of the symbols than the standard VQ were considered. Concatenation of a lossless compression scheme with VQ makes the overall system a variable rate codec. It was reported that concatenation of an appropriate lossless coding technique with a structured VQ system gave improved rate–distortion performance for image and video signals but only marginally better performance for the VQ of linear prediction parameters derived from speech.

In this chapter, a novel dynamic codebook re-ordering (DCR) procedure is described in which the codebook of standard VQ is re-ordered for every encoded symbol based on a suitably chosen dissimilarity measure. The dissimilarity measure chosen for this re-ordering procedure depends only on the codevectors selected in the past. Therefore, the DCR procedure can be replicated at the receiver without requiring any additional transfer of information. The proposed DCR procedure does not introduce any sub-optimality to the VQ system and is therefore capable of achieving a significantly better rate–distortion performance than the standard VQ.

By incorporating the DCR procedure in the standard VQ and compressing the output symbol sequence using a suitable lossless compression system, a fixed rate coding system is converted into a variable rate system. Variable bit-rate encoding is often used in image and video compression standards. For instance, the JPEG 2000 image compression standard [88] employs a binary arithmetic coding scheme to encode the coefficients of the discrete wavelet transform and the MPEG4 standard for video compression uses Huffman coding [83]. The use of lossless compression postprocessor in speech coders in personal and mobile communication systems is also

becoming increasingly popular [17].

The DCR procedure is introduced in Section 6.2. The algorithms for the encoder and the decoder of a VQ system employing the DCR procedure is given in Section 6.2.1 and Section 6.2.2 respectively. Simulation results using a Gauss Markov vector sources are provided in Section 6.3 that demonstrate the entropy reduction achievable using DCR. To illustrate a practical application, the DCR procedure is applied in the VQ of the line spectral frequencies (LSFs) obtained from speech signals in Section 7.1. It is demonstrated that the DCR procedure may be employed to encode the LSFs with fewer bits than the traditional VQ techniques, without any additional loss of information.

6.2 Dynamic codebook re-ordering

Let a sequence of source vectors, $\mathbf{x}[0], \mathbf{x}[1], \ldots, \mathbf{x}[t], \ldots$ be encoded using VQ and let $Q(\mathbf{x}[0]), Q(\mathbf{x}[1]), \ldots, Q(\mathbf{x}[t]), \ldots$ be the corresponding prototype vectors chosen by the vector quantizer. Although we will refer to the independent variable t as the "time instance", it can be given other interpretations and easily extended to sequences of vectors in space, etc. in the following discussions.

In many practical applications, consecutive source vectors are often correlated. In this section, the DCR algorithm is described that exploits the correlation between consecutive source vectors to skew the PMF of the symbols in ζ, thus resulting in a reduction in the entropy H.

The motivation for the DCR algorithm stems from the fact that consecutive source vectors often tend to be vector quantized to the same prototype vector or similar prototype vectors. Thus at each time instance t, the proposed DCR algorithm re-orders the prototype vectors in the codebook in the increasing order of a suitably chosen dissimilarity measure between $Q(\mathbf{x}[t])$ and all other prototype vectors in $\mathcal{C}$. The dissimilarity measure can be designed as a valid distance measure $D(Q(\mathbf{x}[t]), C_k)$,

for $k = 0, 1, \ldots, K$. Since consecutive source vectors are assumed to be correlated, it can be expected that the $Q(\mathbf{x}(t+1))$ is similar to $Q(\mathbf{x}[t])$. In other words, it is likely that index of $Q(\mathbf{x}(t+1))$ in the re-ordered codebook is close to 0. Since the dissimilarity measure used for the re-ordering depends only on the prototype vectors in the codebook, which is available to the decoder, the DCR procedure can be duplicated at the decoder without any additional transfer of information.

To illustrate the DCR with an example, consider a 2 bit codebook, ($K = 4$), with codevectors $\{C_0, C_1, C_2, C_3\}$. At $t = 0$, let the codevector selected corresponding to $\mathbf{x}[t]$ be C_2 and therefore the symbol transmitted is $i(0) = 2$. The DCR procedure is then applied to re-order the codebook in the increasing order of the dissimilarity measure $D(C_2, C_k)$. Let $D(C_2, C_2) < D(C_2, C_3) \leq D(C_2, C_1) \leq D(C_2, C_0)$ so that the re-ordered codebook is $\{C_2, C_3, C_1, C_0\}$. This sorting procedure can be replicated at the decoder too. At $t+1$, if $\mathbf{x}[t]$ and $\mathbf{x}[t+1]$ are correlated, then the codevector chosen corresponding to $\mathbf{x}[t+1]$ is likely to be similar to C_2. Therefore, it can be expected that the symbol chosen by the encoder, $i[t+1]$ is highly likely to be either 0 or 1, the locations of either C_2 or the codevector most similar to C_2. Again the codebook is re-ordered in the increasing order of dissimilarity between the codevector chosen at $t+1$ and all other codevectors. If the consecutive source vectors are correlated, it is evident that the DCR algorithm will result in the transmitted symbols to be close to 0.

Implementation of the proposed DCR by physically reorganizing the codebook, which is typically stored in a memory, will require interchange of the contents of the memory and therefore may be prohibitively expensive. A much more efficient implementation of the DCR procedure can be achieved by employing a simple dynamic index map, $\Psi[k, t]$, that relates the physical address, k, of a prototype vector C_k in the codebook $\mathcal{C}$ to its corresponding re-ordered index at each time instance t. Thus, $\Psi[k, t]$ can be thought of as the index of C_k in the re-ordered codebook at time t.

Unlike in the standard VQ where the digital symbol i at instance t (denoted $i[t]$) is set to the index k, in the VQ scheme employing DCR $i[t]$ is set to $\Psi[k,t]$. In the following two subsections, an algorithmic description the VQ encoder and decoder with DCR is provided.

6.2.1 VQ encoder with DCR

In this subsection, the VQ encoder algorithm that employs the proposed DCR is described. At $t = 0$, the dynamic index map, $\Psi[l, 0]$ is initialized as

$$\Psi[l, 0] = l, \text{ for } l = 0, 1, 2, \ldots, K-1. \tag{87}$$

For $t = 0, 1, 2, \ldots$ the encoding algorithm is given by

1) **Codebook search:** Given the source vector $\mathbf{x}[t]$, the codebook $\mathcal{C}$ is searched according to (83) to determine the "best match" prototype vector C_k. Thus, $Q(\mathbf{x}[t]) = C_k$.

2) **Dynamic index map:** The physical index k corresponding to $\mathbf{x}[t]$ is mapped to the re-ordered index using $\Psi[k,t]$ and the VQ encoder symbol $i[t] = \Psi[k,t]$ is made available to the decoder.

3) **Dynamic codebook re-ordering:** This step updates $\Psi[k,t]$. For $l = 0, 1, 2, \ldots, K-1$, the dissimilarity measure $D(Q(\mathbf{x}[t]), C_l)$ is calculated. Let us denote

$$\delta[l,t] = D(Q(\mathbf{x}[t]), C_l) \text{ for } l = 0, 1, 2, \ldots, K-1. \tag{88}$$

δ is then arranged in an increasing order. Let

$$\delta[l_0,t] \leq \delta[l_1,t] \leq \delta[l_2,t] \leq \ldots \leq \delta[l_K,t] \tag{89}$$

where $l_0, l_1, \ldots, l_K \in \{0, 1, 2, \ldots, K-1\}$. The dynamic index map $\Psi[j, t+1]$ is determined according to

$$\Psi[j,t+1] = l_j \text{ for } j = 0, 1, 2, \ldots, K-1. \tag{90}$$

It may be noted that since $\mathbf{x}[t]$ was vector quantized to C_k, $\Psi[k, t+1] = 0$ and the prototype vectors most similar to C_k have a corresponding dynamic index map that is close to 0.

Since correlated source vectors can be expected to be vector quantized to similar prototype vectors, the VQ encoder symbol $i[t] \equiv \Psi[k, t]$ frequently assumes values close to 0. The PMF of the VQ encoder symbol is largely skewed towards values closer to 0.

6.2.2 VQ decoder with DCR

Similar to the decoder, the encoder initializes its dynamic index map according to (87). For $t = 0, 1, 2, \ldots$

1) **Inverse dynamic index map:** The encoder makes the symbol $i[t]$ available to the decoder. Since $\Psi[k, t]$ at t is injective, the physical address (index), k can be obtained from $i[t]$ through inverse dynamic index map.

$$k = \Psi^{-1}[i[t], t] \tag{91}$$

2) **Reconstruction** The decoder then reconstructs $\mathbf{x}[t]$ as C_k.

3) **Update dynamic index map** Since $Q(\mathbf{x}[t])$ is known at the decoder, $\Psi[j, t+1]$ for $j = 0, 1, 2, \ldots K-1$ is determined similar to the encoder (Dynamic codebook re-ordering step in the encoder description).

6.2.3 Extensions to DCR

To exploit correlations that extend beyond the previous vector, the DCR procedure may be generalized by designing the dissimilarity measure to include previously selected prototype vectors, i.e., $Q(\mathbf{x}[t-1]), Q(\mathbf{x}[t-2]), \ldots]$. One of the disadvantages of incorporating the DCR procedure is increase in the computational complexity

of the encoding and the decoding algorithms. This may be alleviated by performing the DCR procedure less frequently, say every other instance, depending on the constraints of the specific system under consideration. Also, since the dissimilarity measure depends only on the contents of the codebook, $D(C_k, C_l)$ for all $k, l \in \zeta$ can be pre-calculated. Thus for each prototype vector C_k in the codebook, a list of indices of prototype vectors in the increasing order of their dissimilarity from C_k can be computed before hand and stored in a memory. The dynamic index map $\Psi[k, t]$ can then be easily derived by a simple lookup of this memory. Depending on the constraints of the application at hand, the list of indices can be pruned.

6.3 DCR in the VQ of Gauss Markov sources

In this section the entropy reduction achievable by employing the proposed DCR technique in the VQ of first order Gauss Markov vector sources is analyzed. Assume that $\mathbf{x}[t]$ is a Gauss Markov vector source characterized by

$$\mathbf{x}(t+1) = \beta \mathbf{x}[t] + \mathbf{v}[t] \tag{92}$$

where $\beta \in (-1, 1)$ is the correlation parameter and $v[t]$ is an N dimensional vector of unit variance IID Gaussian random variables. This type of correlation between successive vectors can be found in many commonly encountered signals like speech, video and image, etc. Thus the results presented below are extendable to all such applications.

The dimension, N, of $\mathbf{x}[t]$ is chosen to be 10. To analyze the performance of the proposed DCR procedure, several experiments were performed with different codebook sizes K and Gauss Markov sources with different correlation parameters (β). In each of these experiments, a training database of 100000 vectors was generated according to (92), with the appropriate β. The K prototype vectors in the codebook were trained using the generalized Lloyd's algorithm [35]. The Euclidian distance

was used as the distortion measure in (83). A testing database of 20000 vectors (outside the training database) was created corresponding to each β. The VQ encoding and decoding with DCR were performed on the testing database. The dissimilarity measure $D(Q(\mathbf{x}[t]), C_l)$ employed in the DCR algorithm is Euclidian distance,

$$D(Q(\mathbf{x}[t]), C_l) = \|Q(\mathbf{x}[t]) - C_l\|^2. \tag{93}$$

For each experiment, the VQ encoded symbols were recorded and their PMF, was calculated. It may be recalled that the VQ encoded symbols are set to the dynamic index map $\Psi[k, t]$ when DCR is used in the VQ process.

The entropy H corresponding to a PMF, $p[i]$, is calculated according to (85). Figure 13(a) shows the PMF of the symbols of a vector quantizer with 4096 prototype vectors in its codebook ($K = 4096$) for a Gauss Markov source $\mathbf{x}[t]$ with $\beta = 0.9$ when the proposed DCR is not employed. This corresponds to the standard implementation of vector quantization. It must be noted that in this case distribution of the VQ encoder symbols is fairly uniform. The entropy of this system is found to be approximately 12 bits. Figure 13(b) shows the PMF of the VQ encoder symbols of a 12 bit vector quantizer when the proposed re-ordering is employed. In this case, the symbols close to 0 occur more frequently than symbols further away from 0. The entropy of this system is found to be 7.3. Similar PMF plots for a Gauss Markov source $\mathbf{x}[t]$ with $\beta = 0.3$ are shown in Fig. 13(c) and (d). The entropy corresponding to the PMF of the VQ encoding symbols when DCR is employed (Fig. 13(c)) is found to be 11.4. From Fig. 13 (b) and (d), we may conclude that the PMF of the VQ encoding symbols with DCR is more skewed (higher probabilities of the symbols being close to 0) when the correlation in the Gauss Markov source is higher than when the correlation parameter is lower.

The percentage reduction in Entropy ($\Delta H(\%)$) achieved by employing the proposed DCR in VQ of Gauss Markov sources with different correlation parameters is shown in Fig. 14. For a vector quantizer designed with K prototype codevectors,

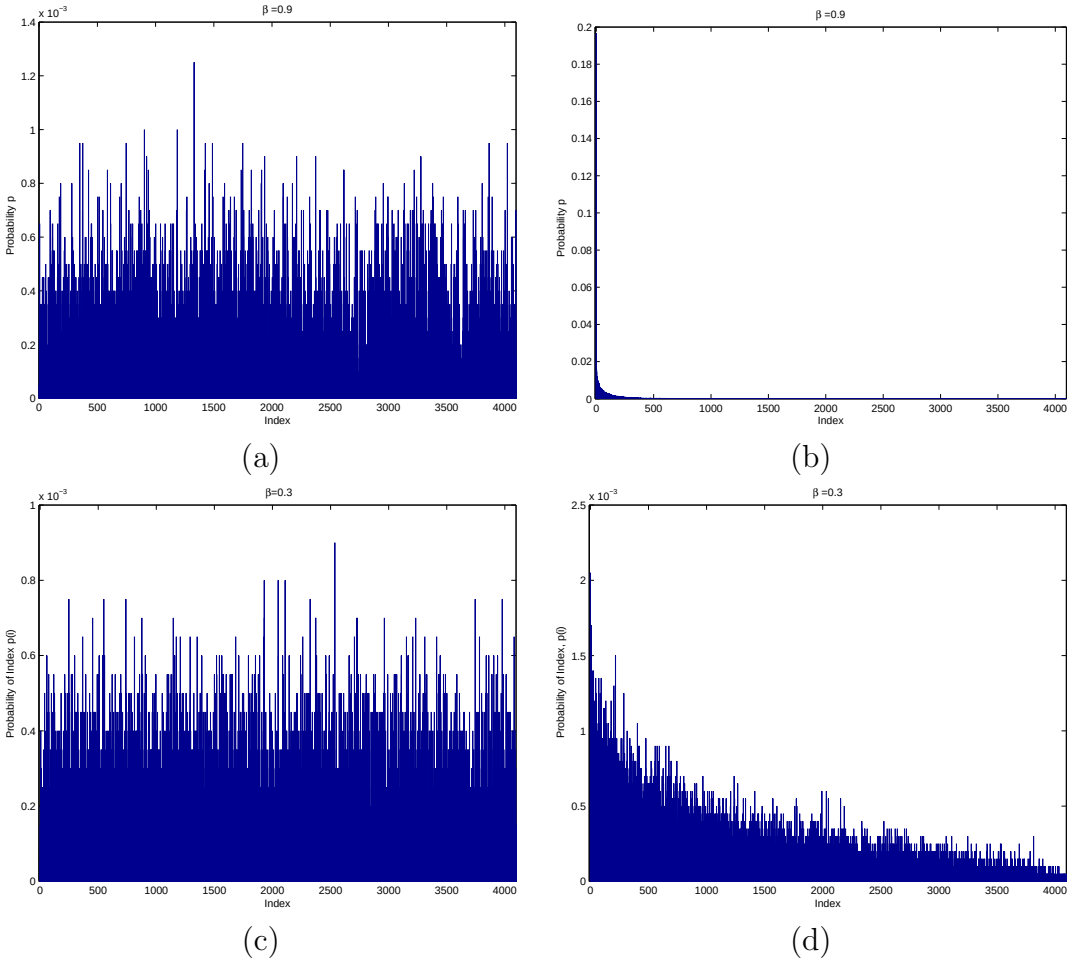

Figure 13. (a) PMF of the symbols of a VQ without DCR (b) PMF of the symbols of a VQ employing DCR for a Gauss Markov vector source with $\beta = 0.9$. (c) PMF of the symbols of a VQ without DCR (d) PMF of the symbols of a VQ employing DCR for $\beta = 0.3$

$\Delta H(\%)$ is defined as

$$\Delta H(\%) = \frac{H_{reorg} - \log_2 K}{\log_2 K} \times 100 \tag{94}$$

where H_{reorg} is the entropy in the PMF of the VQ encoding symbols when the proposed DCR is employed (85). As expected, larger $\Delta H(\%)$ is obtained when the correlation parameter is higher. Also, in most cases, it is observed that $\Delta H(\%)$ is higher for larger codebook sizes (larger K). To demonstrate the effect of the vector dimension, N, on $\Delta H(\%)$, 5 vector quantizers were designed for Gauss Markov vector sources of 5 different dimensions ($N = 2, 4, 6, 8$, and 10) and correlation parameter

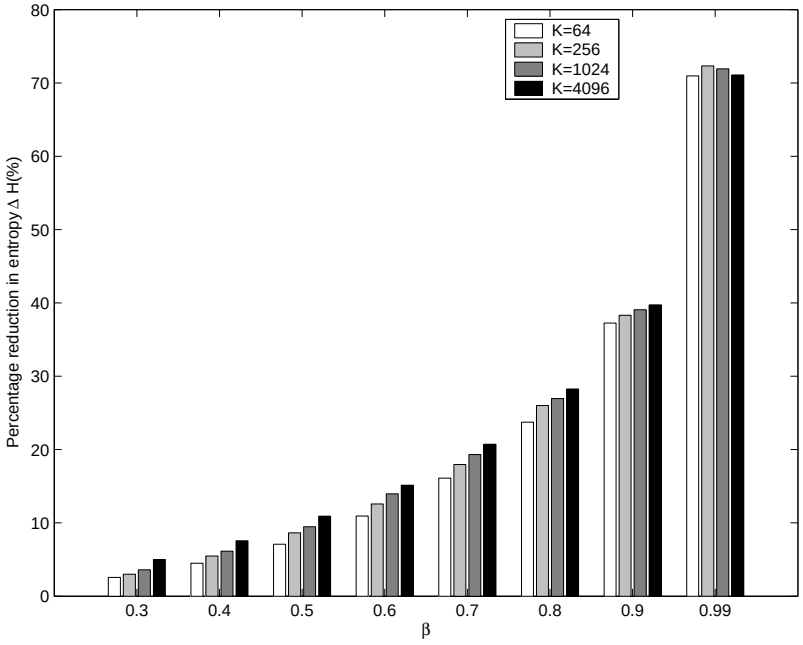

Figure 14. Percentage reduction in entropy achieved when the proposed DCR algorithm is employed in the VQ of Gauss Markov sources. β is the correlation parameter and K is the size of the VQ codebook.

	Vector Dimension N				
	2	4	6	8	10
$\Delta H(\%)$	40.8	38.84	38.81	39.09	39.07

Table 19. Variation of $\Delta H(\%)$ with the dimension of the vectors N

$\beta = 0.9$. In each case, K was chosen such that $\frac{\log_2(K)}{N} = 1$. This ensures that all five vector quantizers have the same resolution. The results are presented in Table 19. We may infer that $\Delta H(\%)$ is approximately the same in all five cases.

6.4 Summary

In this paper, we developed and presented a dynamic codebook re-ordering procedure that enables a reduction in the number of bits required to encode the output symbols of a VQ system for correlated source vectors. The proposed DCR reduced the entropy of the output symbols of the VQ encoder by exploiting intervector correlation. The effectiveness of the DCR procedure arises from the fact that the entropy reduction is

achieved without any increase in distortion as compared to a standard VQ system. While incorporating the DCR in the encoder and the decoder of the VQ will result in an increase in the complexity, some of it can be alleviated by pre-calculating the re-ordering or by performing a partial re-ordering. Also the proposed DCR procedure does not increase the algorithmic delay in either the encoder or the decoder. Therefore, the proposed DCR algorithm can profoundly impact several variable rate signal coding systems by significantly improving their rate–distortion performance. In the next chapter, the incorporation of the DCR procedure in the VQ of parameters of the MELP coder will be discussed

CHAPTER 7

DYNAMIC CODEBOOK RE-ORDERING FOR VARIABLE BIT-RATE MELP CODING

It is well known that parameters of the MELP derived from consecutive segments of a speech signal are strongly correlated. Traditionally, several techniques including structured vector quantization (VQ) [35], super-frame VQ [98] [97] and VQ with memory [35] have been proposed for reduced bit-rate coding of the speech coder parameters. However, many of these techniques render the coding process sub-optimal, or require buffering and thus introduce coding delay or introduce distortions in the reconstructed vector.

In Chapter 6, we presented a dynamic codebook re-ordering (DCR) procedure that when incorporated into a standard VQ system resulted in a considerable reduction in the entropy of the VQ encoder symbols. The DCR procedure effectively exploits the correlation between consecutive vectors without introducing any delay, distortion or sub-optimality to the standard VQ system. In this chapter, we apply the DCR procedure to the VQ of the parameters of a MELP vocoder. It is demonstrated that the DCR procedure may be employed to significantly reduce the entropy of the symbols of the encoders that are used to encode the parameters of the MELP vocoder. This reduction in entropy can be translated into a reduction in the average transmitted bit-rates by employing a suitable lossless compression scheme.

The DCR procedure converts a fixed rate VQ system into a variable rate system, if a lossless compression scheme such as Huffman coding [83] is applied to the encoder output symbols. Variable rate speech coders have become increasingly popular in personal and mobile communication systems and in voice over IP (VoIP) applications. A brief survey of the state of the art variable rate coders and their impact on these voice communication technologies were presented in Chapter 2, Section 2.1.3. The

flexibility offered by the variable rate coders in utilizing the available bandwidth as needed and in enabling more efficient error control mechanisms have resulted in increase in the capacity of mobile and VoIP systems to handle more voice data.

It may be recalled that the parameters of the MELP coder include the LSFs, bandpass voicing constants (BPVC), pitch, gain, aperiodic flag and the fourier magnitudes [2]. In the following sections, implementation of the DCR procedure in the quantization of each of the parameters (excluding the aperiodic flag, which is a single bit every frame) is described.

7.1 DCR in VQ of line spectral frequencies

Most state of the art, low bit-rate speech coders represent the speech signal as the output of an autoregressive (AR) model. The parameters of the AR model are derived from short duration (10-30 msec) segments of the speech signal by a procedure called linear prediction (LP) analysis. To enable the reconstruction of a stable AR model at the receiver, speech coders typically convert the model parameters into line spectral frequencies (LSFs) and VQ is often used to encode the LSFs [49].

In practical speech coders, typically, a vector of 8–12 LP parameters derived from an appropriately windowed segment of speech will have to be coded with at least 24 bits to maintain good perceptual quality of the reconstructed segment [86] [74]. Thus, an unconstrained optimal vector quantizer with 2^{24} prototype vectors in its codebook will be required to encode these LP parameters. This renders the encoding complexity and the storage requirements prohibitively large.

Several structurally constrained VQ techniques reduce the complexity of implementation for a marginal degradation in the reconstruction quality compared to the optimal VQ. In general, structurally constrained vector quantizers encode a source vector $\mathbf{x}$ using a set of K encoders $(\mathbf{Q}_1, \mathbf{Q}_2, ..., \mathbf{Q}_K)$ and correspondingly K

decoders,$(\widehat{\mathbf{Q}}_1, \widehat{\mathbf{Q}}_2, ..., \widehat{\mathbf{Q}}_K)$,that are arranged in a predetermined architecture. Associated with each of the VQ encoder–decoder pair, $\mathbf{Q}_i, \widehat{\mathbf{Q}}_i$ is a codebook $\mathbf{C}^{(i)}$. In a multistage VQ (MSVQ) system, these K encoders are arranged in a cascade such that $\mathbf{x}$ is reconstructed as the sum of the appropriately chosen prototype vectors from these codebooks. In a split VQ system, the N dimensional input vectors are split into K smaller sub-vectors and the encoder–decoder pairs operate in parallel and independently on each of these sub-vectors.

In this section, we describe the incorporation of the DCR procedure in structured vector quantizers. This formulation will be general and applicable to both MSVQ and split VQ systems. Implementation details for these two specific cases will be described in Sections 7.1.1 and 7.1.2, respectively.

Let the number of codevectors in the i^{th} codebook of the structured VQ be $n^{(i)}$. In the encoding mode, let $\mathbf{x}(t)$ be the LSF vector at a given time instance t. Assume that the prototype vectors selected to encode $\mathbf{x}(t)$ are $\{\mathbf{C}^{(1)}_{j_1}, \mathbf{C}^{(2)}_{j_2}, \ldots \mathbf{C}^{(K)}_{j_K}\}$ where $j_i \in \{0, 1, \ldots, n^{(i)} - 1\}$. Also let the symbols transmitted corresponding to these codevectors at t be denoted $\varsigma^{(1)}[t], \varsigma^{(2)}[t], \ldots \varsigma^{(K)}[t]$ respectively. In a standard structurally constrained VQ, without DCR, $\varsigma^{(i)}[t] \equiv j_i$.

The DCR algorithm described in Section 6.2 is applied to each codebook $\mathbf{C}^{(i)}$, $i = 1, 2, \ldots, K$, at each instance t, depending on the prototype vector selected from that codebook. A unique dynamic index map (refer Section 6.2), $\Psi^{(i)}(l_i, t)$ is defined for each of the K encoder–decoder pair and is initialized according to $\Psi^{(i)}[l_i, 0] = l_i$ for $l_i \in \{0, 1, \ldots, n^{(i)} - 1\}$. At time instance t, the codebook $\mathbf{C}^{(i)}$ is re-ordered at the encoder and decoder according to the procedure described in Section 6.2.1 and 6.2.2 respectively, using the dissimilarity measure, $D(\mathbf{C}^{(i)}_{j_i}, \mathbf{C}^{(i)}_{l_i})$ for $l_i \in \{0, 1, \ldots, n^{(i)} - 1\}$. At t, the set of transmitted symbols corresponding to the k encoders, $\{\varsigma^{(1)}, \varsigma^{(2)}, \ldots, \varsigma^{(K)}\}$ are given by, $\{\Psi^{(1)}[j_1, t], \Psi^{(2)}[j_2, t], \ldots, \Psi^{(K)}[j_K, t]\}$ respectively.

The entropy in the distribution of the symbols generated by the i^{th} encoder can

be obtained similar to (85),

$$H^{(i)} = -\sum_{\text{all symbols } \varsigma^{(i)}} p[\varsigma^{(i)}].\log_2\left(p\left[\varsigma^{(i)}\right]\right). \tag{95}$$

If no DCR is employed in any of the encoder decoder pairs, each $\mathbf{x}[t]$ will be encoded at a rate of $\sum_{i=1}^{K}\log_2 n^{(i)}$ bits. The performance improvement in the structurally constrained VQ when the DCR procedure as described above is quantified as either the percentage reduction in the sum of the K encoder output entropies ($\%\Delta H_S$) or the percentage reduction in the joint entropy ($\%\Delta H_J$). The former is defined by

$$\%\Delta H_S = \frac{\sum_{i=1}^{K}\left(\log_2 n^{(i)} - H^{(i)}\right)}{\sum_{i=1}^{K}\log_2 n^{(i)}} \times 100 \tag{96}$$

The joint entropy in the H_J can be defined in terms of the joint PMF of the K stage symbols denoted by, $p(\varsigma^{(1)}, \varsigma^{(2)}, \ldots, \varsigma^{(K)})$.

$$H_J = -\sum_{\text{all possible } \{\varsigma^{(1)},\varsigma^{(2)},\ldots,\varsigma^{(K)}\}} p(\varsigma^{(1)}, \varsigma^{(2)}, \ldots, \varsigma^{(K)})\log_2\left(p(\varsigma^{(1)}, \varsigma^{(2)}, \ldots, \varsigma^{(K)})\right). \tag{97}$$

Therefore the percentage reduction in the joint entropy ($\%\Delta H_J$) is given by

$$\%\Delta H_J = \frac{\sum_{i=1}^{K}\log_2 n^{(i)} - H_J}{\sum_{i=1}^{K}\log_2 n^{(i)}} \times 100 \tag{98}$$

7.1.1 DCR for MSVQ

Multistage vector quantizers (MSVQ) are among the most popular structurally constrained vector quantizers used in speech coding applications. In [57], it was reported that transparent coding of speech can be achieved with approximately 20-24 bit, 2-4 stage jointly designed tree structured MSVQ. A detailed discussion of MSVQ, including a description of the joint codebook design procedure and M search algorithm [5] for improved coding is provided in [57].

To illustrate the improvement in performance when DCR is incorporated in the encoding and decoding process of MSVQ, two stage ($K = 2$) MSVQ encoders and decoders were jointly designed for a range of total bits $\left(\log_2 n^{(1)} + \log_2 n^{(2)}\right)$ from 16

to 24 bits. 100000 consecutive 10-dimensional LSF vectors were derived from 37.5 minutes of speech using 180 samples segments. These vectors were then encoded by the MSVQ with DCR applied to both the stage codebooks. The symbols generated were then used to evaluate the empirical PMFs.

The $\%\Delta H_S$ and $\%\Delta H_J$ for different values of the sizes of the first and the second stage codebooks are shown in Tables 20 and 21, respectively.

$\log_2 n^{(1)}$	$\log_2 n^{(2)}$				
	8	9	10	11	12
8	17.7282	17.1944	16.5032	15.9796	15.2707
9	17.1932	16.7734	16.1079	15.7001	15.2259
10	16.9105	16.7621	15.9564	15.286	14.96
11	16.716	16.3737	16.0932	15.264	14.7873
12	16.973	16.5507	15.7668	15.2218	14.5293

Table 20. Percentage reduction in sum of the two stage encoder output entropies ($\%\Delta H_S$) for a two stage MSVQ. $n^{(1)}$ is the number of prototype vectors in the first stage codebook and $n^{(2)}$ is the number of prototype vectors in the second stage codebook.

$\log_2 n^{(1)}$	$\log_2 n^{(2)}$				
	8	9	10	11	12
8	17.5023	18.4382	19.6816	21.2352	23.1030
9	18.5933	19.8706	21.2639	23.233	25.3319
10	20.0767	21.655	23.2742	25.2956	27.5794
11	21.6097	23.3792	25.519	27.5751	29.9150
12	23.6420	25.8129	27.6904	29.9284	32.2138

Table 21. Percentage reduction in joint entropy ($\%\Delta H_J$) for a two stage MSVQ. $n^{(1)}$ is the number of prototype vectors in the first stage codebook and $n^{(2)}$ is the number of prototype vectors in the second stage codebook.

From the results presented in these tables, it may be observed that a higher reduction in entropy is achievable if the symbol outputs of the two stages are jointly encoded. With DCR, it can be concluded that the symbols generated by the first and the second stage encoders exhibit similar trends in terms of their values being close to 0. This explains the significant gains achieved when these symbols are jointly

encoded. Further, in Table 20, the percentage reduction in entropy is larger when the first stage is coarsely encoded. This happens since, for coarser first stage quantization, the vectors encoded by the second stage retain larger correlation. However, it must be noted that this trend is not observable in the case of reduction in the joint entropy (Table 21).

7.1.2 DCR for split VQ

The use of split VQ in encoding the LSFs was described in [74]. In split VQ, the input LSF vector $\mathbf{x}(t)$ is split into K sub-vectors. Each of these K sub-vectors is then encoded by a different encoder $\mathbf{Q}_i, i = 1, 2, \ldots, K$. At the receiver $\mathbf{x}(t)$ is reconstructed by the concatenation of outputs from the K decoders.

In the experiments described below, split VQ systems, in which the input LSF vector was split into two sub-vectors, were designed for values of $(\log_2 n^{(1)} + \log_2 n^{(2)})$ ranging from 16 to 24. The codebooks were trained was done using 200000 training vectors from the TIMIT training database. The 10-dimensional LSF vectors were split such that the first subvector was 4-dimensional and contained the first 4 LSFs and the second subvector was 6-dimensional and contained the remaining 6 LSFs. In the testing mode, 100000 consecutive 10-dimensional LSF vectors were derived from 37.5 minutes of speech using 180 samples segments. These vectors were then encoded by the split VQ with DCR applied to both the codebooks every time instance t. The symbols generated were then used to evaluate the empirical PMFs.

The $\%\Delta H_S$ and $\%\Delta H_J$ for different values of the sizes of the first and the second codebooks are shown in Tables 22 and 23, respectively.

Results presented in Tables 22 and 23 show trends similar to that observed for MSVQ. Again, it may be concluded that lower bit-rate encoding can be achieved if the symbol outputs from the two codebooks of the split vector quantizer are jointly compressed by a lossless coding system.

$\log_2 n^{(1)}$	$\log_2 n^{(2)}$				
	8	9	10	11	12
8	22.0397	21.578	21.0482	20.6525	19.8136
9	21.4733	21.0602	20.4976	20.0622	19.5165
10	20.7455	20.4729	19.9987	19.6189	18.962
11	20.1339	19.9084	19.3857	19.032	18.4967
12	19.3237	19.0722	18.7853	18.423	17.9635

Table 22. Percentage reduction in sum of the two encoder output entropies ($\%\Delta H_S$) for a split VQ. $n^{(1)}$ is the number of prototype vectors in the first codebook and $n^{(2)}$ is the number of prototype vectors in the second codebook.

$\log_2 n^{(1)}$	$\log_2 n^{(2)}$				
	8	9	10	11	12
8	25.4655	26.0803	26.8641	27.8848	28.8581
9	25.9796	26.8539	27.8179	29.0012	30.3553
10	26.5672	27.7355	28.9764	30.3185	31.7675
11	27.4215	28.8691	30.1521	31.6705	33.3071
12	28.2862	29.8556	31.4848	33.1135	34.8873

Table 23. Percentage reduction in joint entropy ($\%\Delta H_J$) for a split VQ. $n^{(1)}$ is the number of prototype vectors in the first codebook and $n^{(2)}$ is the number of prototype vectors in the second codebook.

7.1.3 Performance comparison

From the results presented in Sections 7.1.1 and 7.1.2, we may conclude that both $\%\Delta H_J$ and $\%\Delta H_S$ are higher in the case of split VQ compared to MSVQ. In the case of a jointly designed MSVQ with M search [5] used in the encoding process, the input vector is approximated as a sum of K prototype vectors derived from the K stage codebooks. While two consecutive source vectors, $\mathbf{x}[t]$ and $\mathbf{x}[t+1]$, may be correlated, the prototype vectors correspondingly selected from the same stage codebook at times t and $t+1$ need not exhibit the same degree of similarity. On the other hand, a split VQ independently operates on sub-vectors derived from a given vector. As a result, greater degree of reduction in both the sum of the individual entropies and the joint entropy is achieved with split VQ as compared to MSVQ.

In Fig. 15 (a) and (b), the percentage reduction in the sum of entropies and the

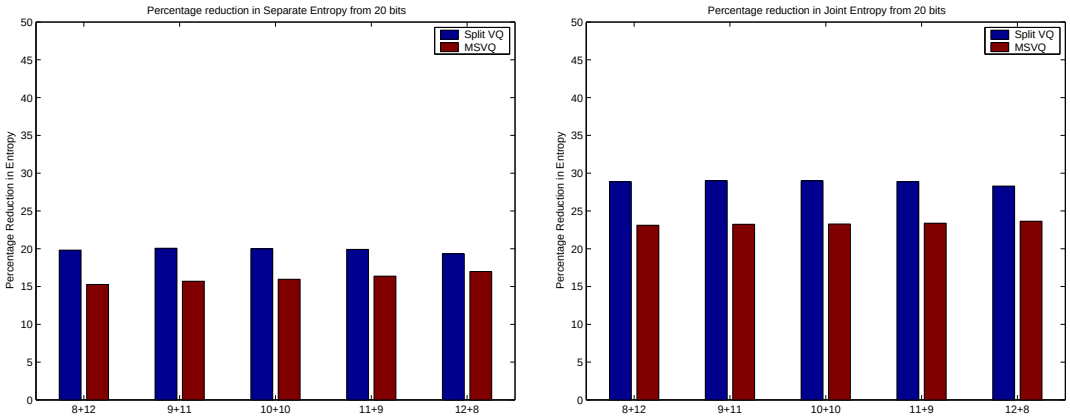

Figure 15. Plots of (a) $\%\Delta H_S$ and (b) $\%\Delta H_J$ for different values of $\log_2 n^{(1)}$ and $\log_2 n^{(2)}$ such that $\log_2 n^{(1)} + \log_2 n^{(2)} = 20$

joint entropy for MSVQ and split VQ for different combinations of the first and the second codebook sizes ($n^{(1)}$ and $n^{(2)}$) are provided. In all these cases, $\log_2 n^{(1)} + \log_2 n^{(2)} = 20$. This would mean that had DCR not been employed in any of the encoder–decoder pairs, each source vector would have been represented by a 20 bit binary symbol.

The choice of the split VQ over the MSVQ is largely due to the higher degree of reduction in $\%\Delta H_S$ and $\%\Delta H_J$ achievable with this architecture. We implement a $K = 2$ split VQ for the 10 LSFs obtained every frame, the first encoder encoding the first 4 LSFs and the second encoder encoding the remaining 6. The codebooks were trained using 200000 LSF vectors obtained from the TIMIT [33] training database. Each of the 2 encoders of the Split VQ uses a codebook with 4096 codevectors. The empirical probability mass function (PMF) of the VQ symbols, derived from 100000 LSF vectors obtained from the speech files in the TIMIT testing database [33], with and without the incorporation of DCR for the split VQ are shown in Fig. 16.

When DCR is not employed, the PMFs of the output symbols of the two encoders are approximately flat. Thus, to encode these symbols, approximately 24 bits are required. With the incorporation of DCR, it is observed that the symbols close to

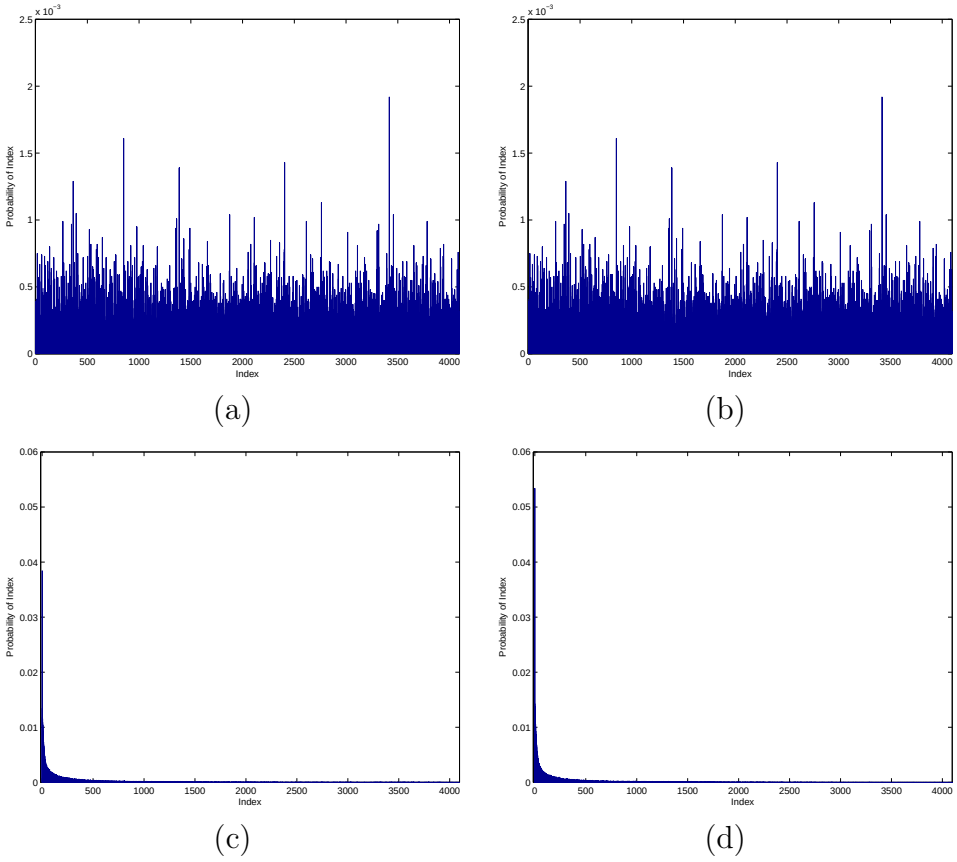

Figure 16. The empirical PMF of the symbol output of the (a) first sub-vector VQ encoder and the (b) second sub-vector VQ encoder without DCR. Correspondingly, (c) and (d) represent empirical PMFs when DCR is employed

0 occur more frequently than symbols further away from 0, thus skewing the PMFs. The empirical entropy corresponding to the empirical joint PMF of the symbols of the two encoders of the split VQ was found to be 16.63.

7.2 DCR in coding the MELP pitch parameter

The pitch parameter, is quantized on a logarithmic scale with a 99-level uniform quantizer ranging from 20 to 160 samples in the MELP coder [2]. This uniform quantizer can be thought of as a 1 dimensional VQ, with the reconstruction levels representing the prototype vectors. The DCR procedure is applied to this quantizer and the empirical PMF of the output symbols of the encoder obtained from 100000 consecutive pitch values is shown in Fig. 17. The resultant empirical entropy was

found to be 3.67 bits.

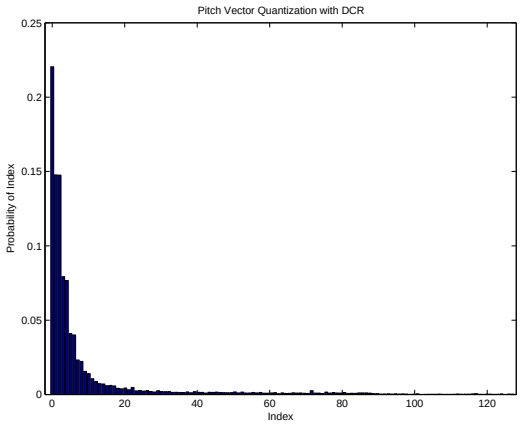

Figure 17. The empirical PMF of the symbol outputs of the uniform scalar quantization encoder with DCR used in *pitch* encoding

7.3 DCR in coding the MELP gain parameter

The two gain values, G_1 and G_2 derived every frame in the MELP coder, are quantized as follows in the standard MELP coder [2]: G_2 is quantized with a 5-bit uniform quantizer ranging from 10 to 77 dB. G_1 is quantized to 3 bits. We replaced the above mentioned encoders with a 2 dimensional vector quantizer with 256 prototype vectors. The quality of the reconstructed speech with the vector quantizer for gain parameters was indistinguishable from the standard MELP reconstruction. The DCR procedure is applied to the encoder and the decoder of the 2 dimensional VQ and the empirical PMF of the output symbols is shown in Fig. 18. The empirical entropy in this case is 6.51 bits.

7.4 DCR in coding the MELP bandpass voicing constants

In the MELP vocoder, the mixed-excitation is implemented using a multi-band mixing model. The auditory spectrum from 0-4 KHz is divided into five bands: 0-0.5 KHz, 0.5-1 KHz, 1-2 KHz, 2-3 KHz and 3-4 KHz. The *voicing decision* in each of these bands is made on the basis of the normalized correlation coefficients of the residuals

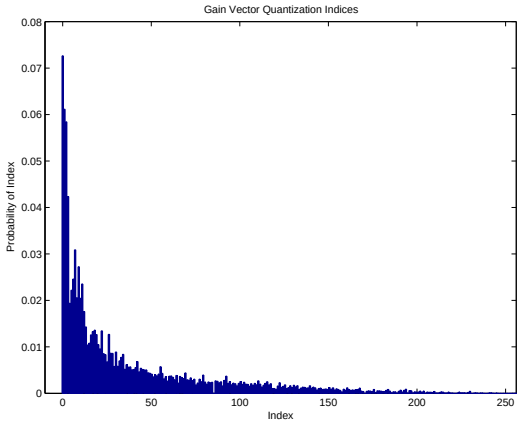

Figure 18. The empirical PMF of the symbol outputs of the 2- dimensional vector quantization encoder with DCR used in *gain* encoding

and the input signal. The encoder conveys this decision to the decoder using 5 bits, each representing the voicing decision of each of the 5 above mentioned bands, respectively.

The 5 bit bandpass voicing decision corresponding to consecutive MELP frames tend to be similar. To exploit this interframe correlation, a dynamic codeword re-ordering procedure is described below that is similar to the DCR.

Thirty-two possible codewords can formed from the 5 bandpass voicing decision bits and these can be stored in a lookup table. This lookup table is similar to the codebook used in the DCR procedure. Instead of using the Euclidian distance to re-order the lookup table, the dissimilarity measure used is the Hamming distance between the codewords. Since 5 codewords exist that are equal hamming distance away from a given 5 bit codeword, codewords that vary in their upper band voicing decisions are assigned lower values in the dynamic index map. As in DCR, the output symbol of the encoder is the mapped codeword.

In Fig. 19, the empirical PMF of the encoder output symbol obtained from 100000 consecutive bandpass voicing decision codewords obtained from 37.5 minutes of speech is shown. It may be noted that several of the 32 codewords never occur since these combinations are not allowed in the MELP coder. The empirical entropy in this case

is 2.6 bits, which is a 48% decrease.

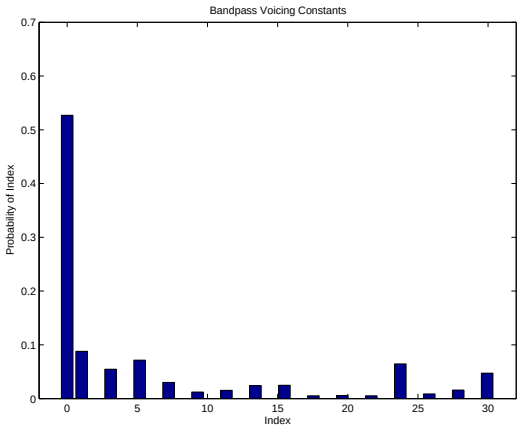

Figure 19. The empirical PMF of the symbol outputs the encoder for *bandpass voicing constants* with DCR

7.5 DCR in coding the MELP fourier magnitudes

Fourier analysis is performed on the LPC residual signal computed using the quantized LPC inverse filter by taking the FFT of an entire frame. At the receiver, the synthesis of each pitch period of the pulse train is done with an inverse DFT of exactly one period in length, using interpolated versions of the transmitted Fourier coefficients for consecutive frames.

The fourier magnitudes are encoded using a 8-bit full-search vector quantizer with bark-scale weighting. The DCR algorithm is incorporated in this VQ procedure. In Fig. 20, the empirical PMF of the encoder output symbol obtained from 100000 consecutive fourier magnitude vectors obtained from 37.5 minutes of speech from the TIMIT testing database is shown. The empirical entropy corresponding to this distribution was found to be 6.06 bits.

7.6 Reduced entropy coding of MELP parameters

With the incorporation of the DCR procedure in encoding the parameters of the MELP coder, it was demonstrated in Sections 7.1–7.3 that the entropy in the output

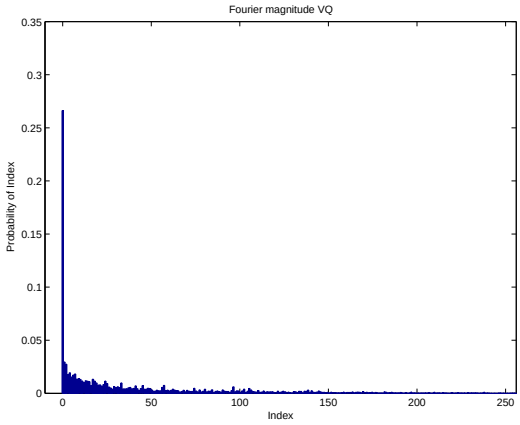

Figure 20. The empirical PMF of the symbol outputs the encoder for *Fourier magnitudes* with DCR

symbols of these coders can be significantly reduced. In Table 24, the entropy in the the encoding the parameters of MELP coder when the DCR employed are tabulated. The number of bits used in the fixed rate coding of the MELP parameters specified in [2] are also compared in Table 24.

Parameter	DOD standard MELP		Entropy: MELP with DCR	
	Voiced	Unvoiced	Voiced	Unvoiced
LSFs	25	25	16.63	16.63
Gain	8	8	6.51	6.51
Pitch	7	7	3.67	0
Bandpass Voicing	5	0	2.60	0
Fourier Magnitudes	8	0	6.06	0
Aperiodic Flag	1	0	1	0
Error Protection	0	13	0	0
Total	54	54	36.47	23.14

Table 24. Bit allocation for MELP coding and empirical entropy in symbols when encoders with DCR are used for MELP parameter coding

In the 100000 frames used in these experiments, 76.94% were voiced and 23.06 % were unvoiced. Therefore, if encoder–decoder pairs with DCR are designed that are capable of representing the parameters using, on an average, the same number of bits as the empirical entropy shown in Table 24, we obtain a coding rate of 1483 bits per second.

7.7 Summary

In this chapter, we demonstrated the application of the dynamic codebook re-ordering algorithm to the encoders and decoders employed in coding the parameters of the MELP speech coder. It was demonstrated that significant reduction in the entropy of the output symbols of these coders can be obtained by the incorporation of the DCR procedure, without any degradation in quality or additional encoding delays compared to the traditional MELP coder. A lossless encoders such as a Huffman coder can be employed to exploit this reduction in entropy. Furthermore, the DCR procedure itself can be designed such that the resultant distribution of the VQ symbols is best suited for a given lossless compression scheme.

CHAPTER 8

JOINT SOURCE CHANNEL CODING FOR ROBUST SPEECH COMMUNICATIONS

The direct use of vector quantization (VQ) to encode LPC parameters of speech in a communication system suffers from the following two limitations: (1) complexity of implementation for large vector dimensions and codebook sizes, and (2) sensitivity to errors in the received indices due to noise in the communication channel. In the past, these issues have been simultaneously addressed by designing channel matched multi-stage vector quantizers (CM-MSVQ) [29]. The CM-MSVQ codec uses a source and channel-dependent distortion measure to encode line spectral frequencies derived from segments of a speech signal. A sub-optimal *sequential* design procedure has been used to train the codebooks of the CM-MSVQ.

In this chapter, we develop and present a channel-optimized multi-stage VQ (CO-MSVQ) codec, in which the stage codebooks are jointly designed [55] [54] [53]. The proposed joint design yields a superior performance in coding the LP parameters as compared to the sequentially designed CM-MSVQ, since each stage codebook is designed to minimize an overall source and channel dependent distortion measure. Each stage encoder of the proposed jointly designed CO-MSVQ codec accounts for the effect of channel errors on the indices generated by all the stage encoders. The M–candidate search procedure described in [5] is used in both the codebook design and the encoding process. Simulation results are provided to demonstrate the improvement in the objective and subjective speech reconstruction quality achieved using the proposed jointly designed CO-MSVQ as compared to CM-MSVQ.

8.1 Channel-optimized VQ of LP parameters

LP analysis is commonly used to obtain the model parameters in the source–system model based speech coding. Although detailed descriptions of LP analysis have been extensively published in the literature, we provide a brief review to introduce the notations that we will use in rest of this chapter. An n^{th} order linear predictor predicts the present sample of the speech signal from a linear combination of the n past samples. If $\hat{s}(m)$ is the predicted present sample, then

$$\hat{s}(m) = \sum_{p=1}^{n} a_p s(m-p). \tag{99}$$

The z domain response of the LP analysis filter is given by

$$A(z) = 1 - \sum_{p=1}^{n} a_p z^{-p}, \tag{100}$$

where $\{a_1, a_2, ...a_n\}$, represent the LP coefficients. The poles of the all-pole synthesis filter correspond to the zeros of $A(z)$. The parameters of the all pole synthesis filter and the excitation signal are encoded and transmitted over the communication channel. At the receiver, the speech signal is reconstructed by filtering the received excitation signal by the the synthesis filter whose response is $1/\widehat{A}(z)$. $\widehat{A}(z)$ is obtained by replacing a_i by the reconstructed LP coefficient at the receiver, $\hat{a}_i$, in Eq. 100. The performance of the encoder in encoding the parameters of the synthesis filter may be quantified in terms of the log spectral distortion (31). To achieve transparent (good quality) reconstruction of the signal, it is necessary that the average SD be less than 1 dB, with less than 2% outliers having a SD more than 2 dB, and no outliers with SD larger than 4 dB [74].

The direct quantization of the LP coefficients may result in an unbounded synthesis filter response. Hence, the n LP coefficients are transformed into a vector of n Line Spectral Frequencies (LSFs), $\mathbf{x} = \{\mathbf{x_1}, \mathbf{x_2}, ...\mathbf{x_n}\}$ that can be efficiently quantized while guaranteeing the stability of the synthesis filter. It has been shown in [49] that

employing VQ to encode the LSPs gives a much better performance, in terms of the objective and subjective reconstruction quality as compared to scalar quantizers.

8.1.1 Optimizing MSVQ for channel characteristics

The performance of a vector quantizer optimized for the characteristics of the signal source alone rapidly degrades in the presence of noise in the communication channel [29]. Under noisy channel conditions, the set of indices generated by the encoders, $I = \{i_1, i_2, ...i_K\}$, is not the same as that received by the corresponding decoders, denoted $J = \{j_1, j_2, ...j_K\}$. This results in a distortion that can be accounted for by including the channel characteristics in the distortion measure used in designing the codebooks of the channel-optimized MSVQ. In general, the transition probability $P(J|I)$ may be used to characterize the channel's statistical properties. The codec, in the presence of channel errors, reconstructs $\mathbf{x}$ as $\sum_{m=1}^{K} \mathbf{C}_{j_m}^{(m)}$ instead of $\sum_{m=1}^{K} \mathbf{C}_{i_m}^{(m)}$. The expected value of the distortion suffered by $\mathbf{x}$ over all possible transitions of I is given by

$$D_{sc}(\mathbf{x}; \mathbf{C}_I) \quad = \quad \sum_{J} P(J|I) \quad \left\|\mathbf{x} - \sum_{m=1}^{K} \mathbf{C}_{j_m}^{(m)}\right\|^2. \tag{101}$$

The subscript "*sc*" indicates that the distortion measure incorporates both the source and the channel characteristics. If the source has an n-fold output probability distribution function $f_{\mathbf{x}}(\mathbf{x})$, then the average distortion $\mathbf{D}$ over all $\mathbf{x}$ in the database, $\mathbf{V}$, is given by

$$\mathbf{D} = \sum_{I} \int_{\mathbf{V}_I} D_{sc}(\mathbf{x}; \mathbf{C}_I) f_{\mathbf{x}}(\mathbf{x}) d\mathbf{x}, \tag{102}$$

where the partition $\mathbf{V}_I$ of the database $\mathbf{V}$ is given by

$$\mathbf{V}_I \quad = \left\{\mathbf{x} : D_{sc}(\mathbf{x}; \mathbf{C}_I) \leq D_{sc}(\mathbf{x}; \mathbf{C}_L) \quad \forall L; \quad L = \{l_1, ...l_K\}\right\}. \tag{103}$$

8.2 Codebook design algorithm for CO-MSVQ

In general, the codebooks of a VQ are designed by using the generalized Lloyd's algorithm [35] to minimize an appropriately chosen distortion measure for a sufficiently rich database of training vectors. The generalized Lloyd's algorithm consists of iteratively partitioning the training vector database into regions for a given set of codevectors and then re-optimizing the codevectors to minimize the distortion over the particular regions.

In a CO-MSVQ, the training database is partitioned according to Eq. 103. For each vector $\mathbf{x}$ in the training database, an optimal (full) search requires considering all possible combinations of codevectors from all stage codebooks and selecting the the set $\mathbf{C}_I$ that minimizes Eq. 102. For the typical codebook sizes encountered in coding LSFs, a full search is computationally expensive. Instead, a suboptimal M–candidate search of the codebooks is employed for a marginal penalty in the reconstruction quality. In the M–candidate search procedure, M codevectors that give the lowest distortion $D_{sc}(\mathbf{x}; \mathbf{C}_{i_1}^{(1)})$ are selected from the first stage codebook. Then, the second stage codebook is searched M times, once for every codevector in the first stage codebook. The M paths that give the lowest overall distortion $D_{sc}(\mathbf{x}; \mathbf{C}_{i_1}^{(1)} + \mathbf{C}_{i_2}^{(2)})$, are selected. This procedure is repeated for all K stages. In the following subsection, we briefly describe the stage-by-stage (sequential) codebook design algorithm presented in [78]. In Section 8.2.2, a joint design algorithm, instead of the conventional sequential design, for training of the CO-MSVQ is proposed.

8.2.1 Stage–by–stage codebook design

In the stage–by–stage codebook design procedure described in [78], the k^{th} stage codebook is designed after the codebooks corresponding to all previous stages (1 to $k-1$) are designed. In designing the k^{th} stage codebook, the partition of the training database is determined by setting $\mathbf{C}_{j_m}^{(m)}$ for $m = k+1, ..., K$, to zero vectors. The $\mathbf{C}_{j_m}^{(m)}$ corresponding to $m = 1, ..., k-1$ are determined from the already designed

codebooks $\mathcal{C}^{(1)}, ..., \mathcal{C}^{(k-1)}$, respectively. The k^{th} stage codevectors are then updated in order to satisfy Eq. 104. After the iterative procedure for designing the k^{th} stage is completed, the subsequent stages are designed. It must be noted that such a sequential design procedure for MSVQ is suboptimal since, while designing a given stage, it assumes that the subsequent stages are populated by zero vectors. In other words, the distortion function that is minimized while designing a given stage, disregards the contribution of all the following stages.

8.2.2 Joint CO-MSVQ codebook design

In the proposed joint CO-MSVQ codebook design algorithm, the codebook corresponding to each stage is designed by incorporating the contributions of all preceding and succeeding stages. Thus in every iteration of the design algorithm, the codebook corresponding to each stage is optimized in order to minimize an overall distortion function (Eq. 102).

The K encoders of CO-MSVQ are jointly designed by using a suitably modified version of the generalized Lloyd's algorithm [35] on a training database of vectors as described below. Each iteration of the design algorithm consists of the following two steps: (1) determining the partition $\mathbf{V}_I$ (Eq. 103) of the database $\mathbf{V}$ for all I, and (2) updating the codevectors, $\mathbf{C}_{j_k}^{(k)}$, of each stage codebook (i.e, $k = 1, 2, ...K$) corresponding to the partitions obtained in step (1), so that the overall distortion of the entire codec, $\mathbf{D}$ is minimized. This can be achieved by setting

$$\nabla_{\mathbf{C}_{j_k}^{(k)}} \mathbf{D} = 0. \tag{104}$$

In general, the squared Euclidian distance is defined as

$$\|\mathbf{x} - \mathbf{y}\|^2 = \big(\mathbf{x} - \mathbf{y}\big)^T \mathbf{W} \big(\mathbf{x} - \mathbf{y}\big), \tag{105}$$

where $\mathbf{W}$ is a positive–definite diagonal matrix. The explicit solution of Eq. 104 can be readily obtained by plugging in the value for $\mathbf{D}$ from Eq. 102 as shown below.

For a given partition $\mathbf{V}_I$ (Eq. 103 of the database of vectors $\mathbf{V}$, the codevectors in the codebook are updated to satisfy Eq. 104. Substituting Eq. 101 and Eq. 102 into Eq. 104,

$$\mathbf{D} = \sum_I \int_{\mathbf{V}_I} \left[\sum_J P(J|I) \| \mathbf{x} - \sum_{m=1}^{K} \mathbf{C}_{j_m}^{(m)} \|^2 \right] f_{\mathbf{x}}(\mathbf{x}) d\mathbf{x} \tag{106}$$

The gradient of $\mathbf{D}$ with respect to $\mathbf{C}_{j_k}^{(k)}$ is given by,

$$\nabla_{\mathbf{C}_{j_k}^{(k)}} \mathbf{D} = \sum_I \int_{\mathbf{V}_I} \sum_{J-\{j_k\}} \left[P(J|I) \quad \nabla_{\mathbf{C}_{j_k}^{(k)}} \| \mathbf{x} - \sum_{m=1}^{K} \mathbf{C}_{j_m}^{(m)} \|^2 \right] f_{\mathbf{x}}(\mathbf{x}) d\mathbf{x} \tag{107}$$

Rewriting the squared Euclidian distance in terms of the product of vectors,

$$\| \mathbf{x} - \sum_{m=1}^{K} \mathbf{C}_{j_m}^{(m)} \|^2 = \left(\mathbf{x} - \mathbf{C}_{j_k}^{(k)} - \sum_{\substack{m=1 \\ m \neq k}}^{K} \mathbf{C}_{j_m}^{(m)} \right)^T \mathbf{W} \left(\mathbf{x} - \mathbf{C}_{j_k}^{(k)} - \sum_{\substack{m=1 \\ m \neq k}}^{K} \mathbf{C}_{j_m}^{(m)} \right) \tag{108}$$

we have

$$\nabla_{\mathbf{C}_{j_k}^{(k)}} \| \mathbf{x} - \sum_{m=1}^{K} \mathbf{C}_{j_m}^{(m)} \|^2 = -2\mathbf{W} \left(\mathbf{x} - \mathbf{C}_{j_k}^{(k)} - \sum_{\substack{m=1 \\ m \neq k}}^{K} \mathbf{C}_{j_m}^{(m)} \right) \tag{109}$$

Substituting (109) into (107), and equating the gradient to 0,

$$\nabla_{\mathbf{C}_{j_k}^{(k)}} \mathbf{D} = -\sum_I \int_{\mathbf{V}_I} \left[\sum_{J-\{j_k\}} P(J|I) 2\mathbf{W} \left(\mathbf{x} - \mathbf{C}_{j_k}^{(k)} - \sum_{\substack{m=1 \\ m \neq k}}^{K} \mathbf{C}_{j_m}^{(m)} \right) \right] f_{\mathbf{x}}(\mathbf{x}) d\mathbf{x} = 0 \tag{110}$$

Rearranging the terms,

$$\sum_I \int_{\mathbf{V}_I} \sum_{J-\{j_k\}} \left[P(J|I) \mathbf{W} \left(\mathbf{C}_{j_k}^{(k)} \right) \right] f_{\mathbf{x}}(\mathbf{x}) d\mathbf{x} = \sum_I \int_{\mathbf{V}_I} \sum_{J-\{j_k\}} \left[P(J|I) \mathbf{W} \left(\mathbf{x} - \sum_{\substack{m=1 \\ m \neq k}}^{K} \mathbf{C}_{j_m}^{(m)} \right) \right] f_{\mathbf{x}}(\mathbf{x}) d\mathbf{x} \tag{111}$$

Thus, the codevector $\mathbf{C}_{j_k}^{(k)}$ is updated according to,

$$\mathbf{C}_{j_k}^{(k)} = \left[\sum_I \int_{\mathbf{V}_I} \mathbf{W} \left\{ \sum_{J-\{j_k\}} P(J|I) \right\} f_{\mathbf{x}}(\mathbf{x}) d\mathbf{x} \right]^{-1} \left[\sum_I \int_{\mathbf{V}_I} \mathbf{W} \, \hat{\mathbf{x}}_k \, f_{\mathbf{x}}(\mathbf{x}) d\mathbf{x} \right], \tag{112}$$

where the expected value of the residual input to the k^{th} stage, $\hat{\mathbf{x}}_k$ is given by

$$\hat{\mathbf{x}}_k = \sum_{J-\{j_k\}} \left[P(J|I) \left(\mathbf{x} - \sum_{\substack{m=1 \\ m \neq k}}^{K} \mathbf{C}_{j_m}^{(m)} \right) \right]. \tag{113}$$

To initiate the iterative codebook training algorithm, an appropriate initial set of the codebooks is required. The initial set of codebooks for the proposed joint codebook design algorithm is obtained by training a portion of the database using the stage–by–stage CM-MSVQ described in [78]. The joint codebook design algorithm is summarized below. Since the partitions and the stage codebooks are modified every iteration, the iteration number t is explicitly included as (t).

1. *Initialization*: The iteration number, t is set to 1. The initial codebooks are set to $\mathcal{C}(0) = \mathcal{C}^{(1)}(0), ..., \mathcal{C}^{(K)}(0)$. k is initialized to 1.

2. *Partition of the training set*: Using the latest set of codebooks, $\mathcal{C}(t-1)$, all partitions of the database are determined (Eq. 103). This step associates a set of indices I with every training vector $\mathbf{x}$. The M–candidate search procedure with an appropriate value of M is employed in evaluating Eq. 103.

3. *Termination criterion check*: The average distortion function for iteration t, $\mathbf{D}(t)$, is evaluated. The training is terminated if $|\mathbf{D}(t) - \mathbf{D}(t-1)|/\mathbf{D}(t)$, drops below a predetermined threshold value, δ.

4. *Codebook update*: The j_k^{th} codevector of the k^{th} stage is updated according to Eq. 112. This is done for all codevectors in the k^{th} stage ($j_k = 1, ..., N$).

5. *Repeat*: If $k = K$, k is re-initialized to 1. The iteration count, t, is incremented by 1 and the algorithm loops back to step 2.

8.3 CO-MSVQ codec operation

In this section, we describe the operation of the CO-MSVQ encoder–decoder pair whose stage codebooks are designed jointly as proposed in Section 8.2.2.

Corresponding to each vector $\mathbf{x}$, the K stage encoders of the CO-MSVQ determine a set of indices I that minimize the distortion function Eq. 101. In other words, for a given $\mathbf{x}$, the set I is determined so that

$$D_{sc}(\mathbf{x}; \mathbf{C}_I) \quad \leq \quad D_{sc}(\mathbf{x}; \mathbf{C}_L) \quad \forall L; \quad L = \{l_1, ..., l_k\}. \tag{114}$$

It must be noted that each encoder stage of the jointly designed CO-MSVQ accounts for the possible distortions suffered by the set of indices I. It is assumed that the characteristics of the channel are known prior to the encoding process. This is a valid assumption since the encoders in several typical communication systems determine the quality of the channel once the link is established and before the actual voice communication begins.

Again, the M–candidate search algorithm is employed in the implementation of Eq. 114. Recall that in the M–candidate search algorithm, the parameter M represents the number of vectors from the first stage codebooks that are considered in the search procedure. These M codevectors form the originating nodes of the search paths. It has been demonstrated in [57] that the performance of the M–candidate search algorithm for moderate values of M (typically, M = 4 to 8) is very close to that of the optimal full search of all the stage codebooks. The choice of M at the encoder poses a trade-off between the reconstruction quality of the LSF vector $\mathbf{x}$ and the encoding complexity.

The CO-MSVQ decoder receives the set of indices J. The codevector $\mathbf{C}_{j_k}^{(k)}$, corresponding to each $j_k \in J$, is retrieved from the respective codebooks. The vector $\mathbf{x}$ is reconstructed as the sum of the codevectors obtained from all the K stage codebooks.

8.4 Jointly designed CO-MSVQ for LSF quantization

In this section, we describe a three stage ($K = 3$) implementation of the proposed jointly designed CO-MSVQ codec to encode LP parameters. First, the speech signal which is sampled at 8000 Hz is segmented into 10 ms frames using a bank of Hamming windows with 20% overlap. Ten LP coefficients derived are from each frame. These

are then transformed into a 10 dimensional vector of LSFs [52]. In order to encode the speech frames with transparent quality, the LSF vector is vector quantized to 24 bits using the proposed jointly designed CO-MSVQ. Each of the three stage codebook is designed to have 256 codevectors.

The jointly designed CO-MSVQ codebooks are trained using 170,000 LSF vectors obtained from speech records in the TIMIT training database. The perceptual weighing matrix $\mathbf{W}$ given in [52] is used in the squared Euclidian distance measure (Eq. 105). $\mathbf{W}$ accounts for the spectral sensitivity of $A(z)$ to the LSFs. It has been demonstrated in [52] that the spectral sensitivity of the i^{th} LSF is directly dependent on the group delay, D_i, of the ratio filter at that frequency. The group delay of the ratio filter is obtained as a byproduct in the process of transforming the LP coefficients to LSFs. The entries of $\mathbf{W}$ are given by

$$W_{i,j} = \begin{cases} 0 & i \neq j \\ u_i \sqrt{D_i / D_{max}} & D_{crit} \leq D_i \leq D_{max} \\ u_i D_i / \sqrt{1.375 D_{max}} & D_i < D_{crit} \end{cases} \tag{115}$$

where

$$u_i = \begin{cases} 1 & x_i < f_{crit} \\ 1 - \frac{1}{6000}\big(x_i - f_{crit}\big) & f_{crit} \leq x_i \leq \frac{f_s}{2} \end{cases} \tag{116}$$

where $f_{crit} = 1000$ Hz, $D_{crit} = 1.375$ ms, $D_{max} = 20$ ms and the sampling frequency $f_s = 8000$ Hz.

For simplicity of formulation, we assume that the indices in the set I suffer distortion independent of each other. In other words,

$$P(J|I) = \prod_{k=1}^{K} P(j_k|i_k). \tag{117}$$

Further, we assume that the channel is binary symmetric with a known bit error probability q. Thus, if the binary representations of i_k and j_k differ by m bits, then,

$$P(j_k|i_k) = q^m \big(1-q\big)^{M-m} \tag{118}$$

where $M = log_2 N$ is the number of bits in the representation of i_k.

The codebooks of the proposed CO-MSVQ are designed as described in Section 8.2.2. The M–candidate search algorithm is employed to partition the training vector database in each iteration of the training process. The performances of jointly designed CO-MSVQ with different values of the parameter M used in the proposed codebook training algorithm, are compared in Figure 21. For each case, the average SD for different values of M used in the testing is plotted. It may be inferred that the best performance (in terms of the lowest SD) is achieved when the value of M used in training matches the one used in the testing.

The CO-MSVQ codebooks are designed and tested for different bit error rates of the binary symmetric channel. The performance of the proposed jointly designed CO-MSVQ, quantified in terms of the average SD of the reconstructed LSF vectors, is evaluated for 5000 test vectors derived from the TIMIT [33] testing database. The comparison between the average SD of the reconstructed LSF obtained using three stage implementations of (1) SO-MSVQ (2) CM-MSVQ described in [78], and (3) the proposed jointly designed CO-MSVQ, is presented in Table 1. It is apparent that the jointly designed CO-MSVQ codec outperforms the sequentially designed CM-MSVQ by approximately 0.12 dB.

The presence of noise in the channel corrupts the reconstruction of the corresponding LP coefficients. The SD corresponding to the speech reconstructed from thesethese "outlier" vectors is typically larger than 2 dB. The degradation in this reconstructed speech can often be heard as short duration "clicks". The percentage of outliers with more than 2 dB of SD and with more than 4 dB of SD for the three cases are given in Tables 2 and 3 respectively. It is observed that the percentage of outliers is consistently lower in the case of the joint codebook design.

Another objective measure that is related the perceptual quality of the speech is the variance of the spectral distortion. It has been shown in [101] that a coding scheme

that yields a lower variance in the spectral distortion often has a better perceptual quality of the reconstructed speech. The variance of the spectral distortion for (i) SO-MSVQ, (ii) sequentially designed CM-MSVQ and (iii) the proposed jointly designed CO-MSVQ are compared in Table 28. These results indicate that the variance of the SD for the proposed design is significantly lower than the conventional sequentially designed CM-MSVQ.

One of the important concerns while implementing CO-MSVQ codec is its performance when the actual bit error rate in the channel is different from the one for which the CO-MSVQ codebooks are designed. Figure 22 shows the performance of the proposed CO-MSVQ whose codebooks under such channel mismatch conditions. The CO-MSVQ codebooks are designed for a (i) low bit error rate channel (BER= 0.0001) and (ii) a high bit error rate channel (BER= 0.01). In each case, the performance is tested for different actual channel bit error rates. For the sake of comparison, the performance of the SO-MSVQ is also plotted (in dotted lines) in the same figure. From this figure, we conclude that the proposed jointly designed CO-MSVQ is highly robust under channel mismatch conditions.

Informal listening test were conducted wherein the listeners were asked to choose between a record of speech encoded using (i) the sequentially designed CM-MSVQ codec and (ii) the proposed jointly designed CO-MSVQ codec. In both cases, the unquantized prediction residuals were used to synthesize the speech frames. In approximately 90% of the cases, the perceptual quality of the reconstruction using the proposed CO-MSVQ codec was found to be superior to that obtained using the sequentially designed CM-MSVQ.

The encoding complexity of the jointly designed CO-MSVQ is the same as that of the CM-MSVQ for the same values of the parameter M of the M–candidate search. It must be noted that although the CM-MSVQ and the proposed jointly designed CO-MSVQ differ in their codebook training algorithms, their encoders (and decoders)

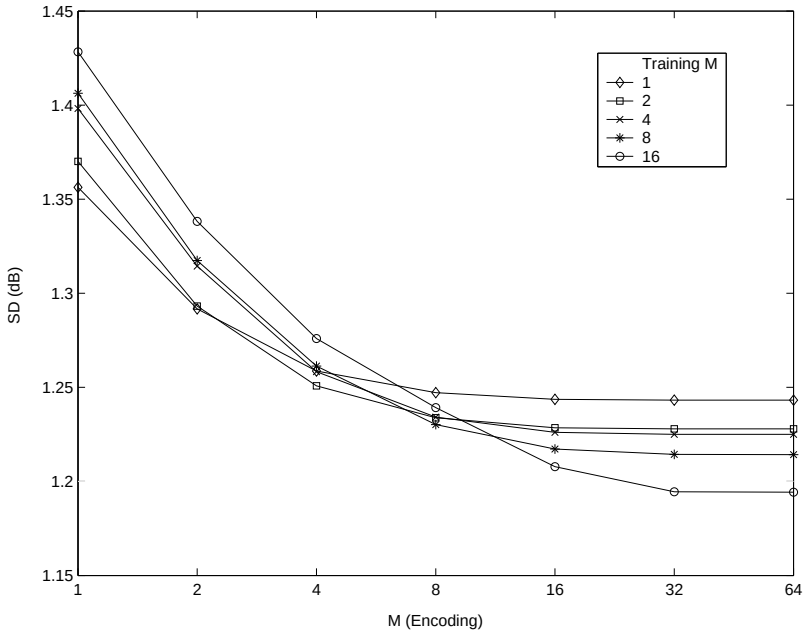

Figure 21. SD (in dB) vs. the value of M used in the encoding process for different values of M used in the training process

are identical. The complexity of the codebook design algorithm is often disregarded since it is done off-line. A detailed analysis of the the encoding complexity of the CM-MSVQ is given in [78].

Bit Error Rate	SO-MSVQ	CM-MSVQ			JD-CO-MSVQ		
		M=1	M=2	M=4	M=1	M=2	M=4
0	1.3785	1.3785	1.3598	1.3134	1.2154	1.2105	1.1905
0.0001	1.4871	1.462	1.408	1.3789	1.3519	1.2884	1.2547
0.001	1.5749	1.5166	1.4596	1.423	1.4219	1.362	1.3175
0.005	2.0034	1.7803	1.7244	1.7187	1.6964	1.6225	1.6161
0.01	2.5439	2.0585	2.0305	2.0176	1.9572	1.929	1.9154

Table 25. Comparison of average SD for 5000 test vectors from TIMIT-test database for three cases: (I) SO-MSVQ, (II) three stage CM-MSVQ, (III) three stage jointly designed-CO-MSVQ

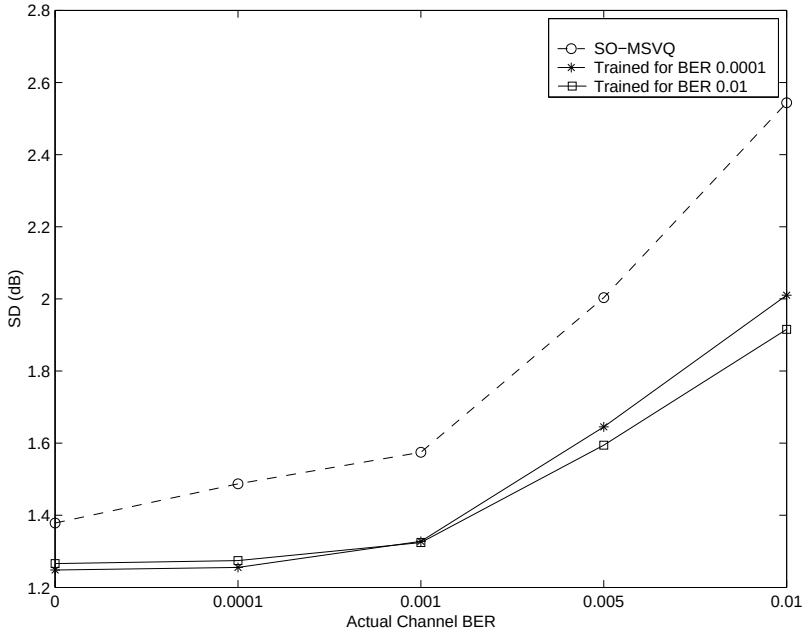

Figure 22. SD (in dB) vs. the value of bit error rate used in the joint design process for CO-MSVQ (i)designed for a bit error rate of 0.0001, (ii) designed for a bit error rate of 0.01 and (iii) SO-MSVQ

8.5 Summary

In this chapter, a channel-optimized multistage vector quantization has been developed and presented. The codebooks of the proposed CO-MSVQ are jointly designed to minimize a source and channel dependent distortion measure. It has been demonstrated that the proposed joint codebook design algorithm, in conjunction with encoding based on M–candidate search, is better suited for a multistage VQ framework than

Bit Error Rate	SO-MSVQ	CM-MSVQ			JD-CO-MSVQ		
		M=1	M=2	M=4	M=1	M=2	M=4
0	7.45	7.45	5.13	4.29	4.45	3.15	2.15
0.0001	14	13.2	11.15	10.33	5.77	3.92	3.37
0.001	15.83	14.37	12.09	11.25	8.74	6.98	6.16
0.005	26.08	21.98	20.47	20.43	16.52	14.39	14.31
0.01	40.21	31.6	30.19	29.95	25.73	24.29	23.85

Table 26. Percentage of outliers with more than 2 dB of SD for 5000 test vectors from TIMIT-test database for three cases: (I) SO-MSVQ, (II) three stage CM-MSVQ, (III) three stage jointly designed-CO-MSVQ

Bit Error Rate	SO-MSVQ	CM-MSVQ			JD-CO-MSVQ		
		M=1	M=2	M=4	M=1	M=2	M=4
0	0	0	0	0	0.002	0	0
0.0001	0.15	0.09	0.13	0.09	0.09	0.09	0.07
0.001	1.27	1.02	0.94	0.86	0.86	0.8	0.76
0.005	5.77	4.18	4.12	3.88	3.94	3.91	3.79
0.01	11.87	10.63	9.59	9.07	9.82	8.45	7.94

Table 27. Percentage of outliers with more than 4 dB of SD for 5000 test vectors from TIMIT-test database for three cases: (I) SO-MSVQ, (II) three stage CM-MSVQ, (III) three stage jointly designed-CO-MSVQ

Bit Error Rate	SO-MSVQ	CM-MSVQ			JD-CO-MSVQ		
		M=1	M=2	M=4	M=1	M=2	M=4
0	0.2538	0.2538	0.2315	0.2315	0.1409	0.1273	0.1228
0.0001	0.8078	0.2654	0.2348	0.3104	0.1772	0.1331	0.2156
0.001	1.1845	0.5848	0.5685	0.4983	0.442	0.4287	0.3695
0.005	4.0154	1.5715	1.5273	1.516	0.8526	0.8431	0.841
0.01	5.1487	2.4897	2.3852	2.287	1.1845	1.1042	1.0953

Table 28. Variance of the SD for 5000 test vectors of TIMIT-Test database for three cases: (I) SO-MSVQ, (II) three stage CM-MSVQ, (III) three stage jointly designed-CO-MSVQ

the conventional sequentially designed CM-MSVQ. The proposed codec is employed to encode the LSFs obtained from speech signal. Simulation results were provided to demonstrate the superior performance of the proposed codec in terms of the average SD relative to the sequentially designed CM-MSVQ. Furthermore, improvements in terms of reductions in the percentage of outliers frames and the variance of the SD have been been demonstrated. It must be noted that this performance improvement is achieved without any additional bit requirements.

APPENDIX A

THE MIXED EXCITAION LINEAR PREDICTION SPEECH CODER

The mixed-excitation linear prediction (MELP) speech coder is based on a technology developed by Dr. Alan McCree and Dr. Thomas P. Barnwell at the Center of Signal and Image Processing, Georgia Institute of Technology, Atlanta. A 2400 kbps version of the MELP algorithm that uses a five-band model and aperiodic pulses was adopted as the new U.S. military standard in 1996 [2]. Recently, an improved version of this coder, which utilizes a better pitch-period estimator and a very high-quality noise suppressor, was also adopted as the new 2.4–1.2 Kbps NATO standard [3]. In this appendix, the description of the as specified in the NATO standard [3] is provided.

A.1 The MELP coding algorithm

The MELP coder is uses an source–autoregressive (AR) system model for representing the speech signal. The AR model parameters are obtained by linear prediction (LP) analysis. Besides the AR model, the 2.4 Kbps MELP includes five additional features to improve the representation of the speech signal. These features include: mixed excitation, aperiodic pulses, an adaptive spectral enhancement filter, a pulse dispersion filter and Fourier series magnitudes. The 2.4 Kbps MELP uses a five band voicing model: 0-0.5 KHz, 0.5-1 KHz, 1-2 KHz, 2-3 KHz and 3-4 KHz. In each of these bands the *voicing decision* is made on the basis of the normalized correlation coefficients of the residuals and the input signal. In addition to the standard pitch detection, a fractional pitch detection algorithm is employed to obtain a better estimate of the *pitch* period within a frame. A 10^{th} order autocorrelation analysis is used to derive the *LPC coefficients* which are then transformed to line spectral frequencies and quantized using a multi-stage VQ. To avoid buzziness in the reconstructed speech

due to a stationary pitch period, the pulse positions are jittered using a *aperiodic flag*. The gain of the signal is calculated as the RMS value of the input signal. The *gain* term is calculated twice for each frame: once at the end of the frame and once in the middle of the frame. As a result, the gain of the signal is accurately represented in this coder. Finally, the magnitudes of the FFT of the residuals are calculated and the first ten harmonics are found by a peak-picking algorithm. The magnitudes of the harmonics are normalized to have an RMS value of 1.0, and then quantized by a 8-bit vector quantizer.

A.2 The 2400 bps MELP encoder

The 2400 bps MELP vocoder encodes the following parameters if a frame of 180 samples of speech is classified as *voiced*: 10 LPC coefficients, pitch, 2 gain values, fourier magnitudes, 5 bit bandpass voicing flag and an aperiodic flag. If the frame is classified as *unvoiced*, the parameters encoded include 10 LPC coefficients, 2 gain values, a 7 bit all zero code for pitch, and error protection. We briefly describe the process of obtaining and encoding these parameters from a frame of speech.

A.2.1 MELP frames

The MELP parameters are estimated using frames obtained every 22.5 ms (180 samples) from the speech signal sampled at 8000 Hz. The last sample in a frame is used as the reference point and all analysis windows used in the estimation of the parameters are centered on this sample. This sample is also referred as the center of the analysis frame. Before the parameter estimation, the input speech is first filtered with a 4th order Chebychev Type-II high-pass filter with a cut-off frequency of 60 Hz. This filter attenuates the low-frequency noise below and around 60 Hz and makes the input signal zero mean. All parameter estimation algorithms use this filtered signal instead of the input speech signal in the encoder. The bit allocation for the MELP parameters for a frame of 180 samples is summarized in Table 29

A.2.2 Linear prediction parameters

A 10th order linear prediction analysis is performed on the input speech signal using a 200 sample (25 ms) Hamming window centered around the last sample in the current frame. The traditional autocorrelation analysis procedure is implemented using the Levinson-Durbin recursion. In addition, a bandwidth expansion coefficient of 0.994 (15 Hz) is applied to the prediction coefficients. The LSF vector is checked for minimum separation of 50 Hz and adjusted accordingly. The resulting LSF vector is then quantized by a multi-stage vector quantizer (MSVQ). The MSVQ codebook consists of four stages whose indices have 7, 6, 6, and 6 bits, respectively.

A.2.3 Bandpass voicing

The input speech is passed through a filter bank that partitions the speech signal into the following five bands: 0-0.5 kHz, 0.5-1 kHz, 1-2 kHz, 2-3 kHz and 3-4 kHz. The normalized correlation coefficients of the ten lags around the initial pitch estimate and those around the initial pitch estimate found in the previous frame are computed using the signal in the 139 lowest band, and the pitch lag with the largest correlation is selected as the frames pitch period. The associated normalized correlation coefficient is called as the voicing (Vbp_1) strength of the lowest band and the voicing strength of the frame. For the remaining bands, the bandpass voicing strengths are determined from the envelope of the filtered outputs of the respective bands. The bandpass voicing strengths are quantized to 1 if their value exceeds 0.6 otherwise they are quantized to 0 for transmission.

A.2.4 Pitch

The initial estimate of the pitch in a frame is obtained by filtering the input signal with a 1 KHz lowpass filter and calculating the autocorrelation of the filtered signal in the 40-160 sample range. Two additional pitch refinements are performed. The first is based on the lowest band in the 5 band bandpass voicing analysis. The second, called

the fractional pitch refinement, This procedure, utilizes an interpolation formula to increase the accuracy of an input pitch value. Details can be found in [3]. Finally, a pitch doubling check procedure is performed that looks for and corrects pitch values which are multiples of the actual pitch.

The final pitch P and the low band voicing strength Vbp_1, are quantized jointly using 7 bits. If $Vbp_1 \leq 0.6$, then the frame is unvoiced and the all-zero code is sent. Otherwise, the log of P is quantized with a 99-level uniform scalar quantizer ranging from $\log_{10} 20$ to $\log_{10} 160$. The resulting index (range 0 to 98) is then mapped to the transmitted 7-bit codeword. The remaining 28 codes with Hamming weight of 1 or 2 are reserved for error protection. This table is also used in decoding the 7-bit pitch code to determine if a frame is voiced, unvoiced, or whether a frame erasure is indicated.

A.2.5 Gain

The input speech signal gain is measured twice per frame using a pitch-adaptive window length. This length is identical for both gain measurements and is determined as follows. When $Vbp_1 \geq 0.6$, the window length is the shortest multiple of P, which is longer than 120 samples. If this length exceeds 320 samples, it is divided by 2. When $Vbp_1 \leq 0.6$, the window length is 120 samples. The gain calculation for the first window produces and is centered 90 samples before the last sample in the current frame. The calculation for the second window produces and is centered on the last sample in the current frame.

A.2.6 Aperiodic flag

The aperiodic flag is set to 1 if $Vbp1 < 0.5$ and set to 0 otherwise. When set, this flag tells the decoder that the pulse component of the excitation should be aperiodic, rather than periodic. The aperiodic flag is a single bit, transmitted as is.

A.2.7 Fourier magnitude

It has been shown in [73] that by including Fourier series magnitudes corresponding to the excitation signal, the quality of the reconstructed speech can be improved significantly. Fourier analysis is performed on the LPC residual signal computed using the quantized LPC inverse filter by taking the FFT of an entire frame. Synthesis of each pitch period of the pulse train is done with an inverse DFT of exactly one period in length, using interpolated versions of the transmitted Fourier coefficients for consecutive frames. The fourier magnitudes are encoded using a 8-bit full-search vector quantizer with bark-scale weighting.

Parameter	DOD standard MELP	
	Voiced	Unvoiced
LSFs	25	25
Gain	8	8
Pitch	7	7
Bandpass Voicing	5	0
Fourier Magnitudes	8	0
Aperiodic Flag	1	0
Error Protection	0	13
Total	54	54

Table 29. Bit allocation for MELP coding

A.3 The 2400 bps MELP decoder

At the decoder, the received symbols are used to retrieve the quantized MELP parameters. When the frame is voiced and the aperiodic flag is set to one, the jitter is set to 25%. Otherwise, it is set to 0%. In unvoiced frames, the pitch period is set to a default 50 samples, the jitter is set to 0% and all Fourier series magnitudes are set to 1.0.

The decoder interpolates the pitch, LSFs, gain, jitter, the band-pass filters used in generation of the mixed excitation and Fourier series magnitudes pitch-synchronously for each synthesized pitch cycle. The interpolation is done linearly between values

these parameters take in the past and current frame based on the starting location of the pitch cycle within the frame. The gain value is interpolated using the first gain and the second gain of the previous frame, when the starting point of the new pitch cycle is before the center of the frame. Otherwise, the first and second gain values are used in interpolation. In the synthesis procedure, first, the starting locations of each new pitch cycle in the frame are found. The pitch-cycle length is computed as the interpolated pitch period plus the contribution due to the jitter factor, which is calculated as the interpolated jitter times a random number uniformly distributed between -1 and 1. The pitch-cycle length is rounded to nearest integer. After the pitch-cycle locations are determined, the first step in the synthesis procedure is to generate the voiced excitation of each new pitch cycle that is computed from the inverse discrete Fourier transform (IDFT) of the interpolated Fourier series magnitudes. After the voiced excitation of the pitch cycle is obtained, the samples in the cycle are multiplied by the square root of the cycle length to obtain a unity RMS signal, and then multiplied by 1000 to obtain a signal with a nominal level. In addition, the samples in the pitch cycle are circularly rotated by ten samples so that the parameter interpolation does not take place at the same location as the pitch pulses with large amplitudes. The noise sequence is generated with a uniform random number generator with an RMS value of 1000. The pulse and noise signals are filtered with the interpolated bandpass filters, and then summed to form the mixed excitation. The adaptive spectral enhancement filter is also generated pitch-synchronously from the interpolated LSFs. The interpolated LSFs are first converted back to the linear-prediction filter, and the adaptive spectral enhancement filter is obtained. Finally, the pulse dispersion filter is applied to the signal continuously.

REFERENCES

[1] "Recommendations p.80 methods of subjective determination of transmission quality." ITU, 1993.

[2] "MIL-STD-3005: Specifications for the analog to digital conversion of voice by 2,400 bit/second mixed excitation linear prediction." Military Standard, US Department of Defense, 1999.

[3] "The 1200 and 2400 bit/s NATO interoperable narrow band voice coder," STANAG no. 4591, NATO Standardization Agency, Den Haag, The Netherlands, 2003.

[4] "Selectable mode vocoder (SMV) service option for wideband spread spectrum communication system," Tech. Rep. 3GPP2 C.S0030-0, 3rd Generation Partnership Project, 2004.

[5] Anderson, J. and Bodie, J., "Least square quantization in PCM," *IEEE Transactions on Information Theory*, vol. IT-21, pp. 379–387, 1975.

[6] Atal, B. and Remde, J., "A new model for LPC excitation for producing natural sounding speech at low bit-rates," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 614–617, 1982.

[7] Atal, B. and Schroeder, J., "Stochastic coding of speech at very low bit-rates," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 1611–1613, 1984.

[8] Babu, B. N. S., "Performance of an FFT based voice coding system in quiet and noisy environments," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-31, p. 1323, Oct. 1983.

[9] Beritelli, F., Casale, S., and Ruggeri, G., "Performance comparison between VBR speech coders for adaptive VoIP applications," *IEEE Communications Letters*, vol. 5, no. 10, pp. 423–425, 2001.

[10] Boll, S. F., "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-27, pp. 113–120, 1979.

[11] Chan, W.-Y., Gupta, S., and Gersho, A., "Enhanced multistage vector quantization by joint codebook design," *IEEE Transactions on Communications*, vol. 40, pp. 1693–1697, Nov. 1992.

[12] Cox, R. V., "New directions in subband coding," *IEEE Transactions on Selected aread in Communications*, vol. 6, pp. 391–409, Feb. 1988. (Special Issue on Voice Coding for Communications).

[13] Cox, R. V. and Crochiere, R., "Real–time simulation of adaptive transform coding," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-29, p. 147, Apr. 1981.

[14] Crochiere, R., "A weighted overlap-add method for short time Fourier analysis/synthesis," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-28, p. 99, Feb. 1980.

[15] Crochiere, R. and Rabiner, L., *Multirate Digital Signal Processing*. Englewood Cliffs, NJ:Printice Hall, 1983.

[16] Cummiskey, P., "Adaptive quantization in differential PCM coding of speech," *Bell systems technical journal*, vol. 52, p. 1105, Sep. 1973.

[17] DeJaco, A., Gardner, W., Jacobs, P., and Lee, C., "QCELP: The North American CDMA digital cellular variable rate speech coding standard," in *IEEE workshop on Speech Coding Telecommunications*, pp. 5–6, 1993.

[18] Deller, J., Proakis, J. G., and Hansen, J. H. L., *Discrete time processing of speech signals*. Englewood Cliffs: Prentice-Hall, 1993.

[19] DeMarca, J. R. B. and Jayanth, N. S., "An algorithm for assigning binary indices to the codevectors of a multidimensional quantizer," in *Proceedings of the IEEE International Communications Conference*, pp. 1128–1132, 1987.

[20] Dempster, A., Laird, N., and Rubin, D., "Maximim likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. 39, pp. 1–38, 1977.

[21] Ephraim, Y., "A minimum mean square error approach for speech enhancement," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 829–832, 1990.

[22] Ephraim, Y., "A bayesian estimation approach for speech signal enhancement using Hidden Markov Models," *IEEE Transactions on Signal Processing*, vol. 40, pp. 725–735, 1992.

[23] Ephraim, Y., "Statistical model based speech enhancement systems," *Proceedings of the IEEE*, vol. 80, no. 10, pp. 1526–1555, 1992.

[24] Ephraim, Y. and Malah, D., "Speech enhancement using a minimum measn squared error spectral amplitude estimator," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-32, pp. 1109–1121, 1984.

[25] Ephraim, Y., Malah, D., and Juang, B.-H., "On the application of Hidden Markov Models for enhancing noisy speech," *IEEE Transactions on Speech and Audio Processing*, vol. 37, no. 12, pp. 1846–1856, 1989.

[26] Ertan, A. E., *Pitch-Synchronous Processing of Speech Signal for Improving the Quality of Low Bit Rate Speech Coders.* PhD thesis, Georgia Institute of Technology, 2004.

[27] Fant, G., *Speech Sounds and Features.* Cambridge, MA: The MIT Press, 1973.

[28] Farvardin, N., "A study of vector quantization for noisy channels," *IEEE Transactions on Information Theory*, vol. 36, no. 4, pp. 799–809, 1990.

[29] Farvardin, N. and Vaishampayan, V., "Optimal quantizer design for noisy channels: an approach to combined source–channel coding," *IEEE Transactions on Information Theory*, vol. IT-33, pp. 827–838, Nov. 1987.

[30] Feansesco, R. D., Lamblin, C., Leguyader, A., and Massaloux, D., "Variable rate speech coding with online segmentation and fast algebraic codes," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 233–236, 1990.

[31] Gadkari, S. and Rose, K., "Unequally protected multistage vector quantization for time varying CDMA channels," *IEEE Transactions on Communications*, vol. 49, no. 6, pp. 1045–1054, 2001.

[32] Gannot, S., Burshtein, D., and Weinstein, E., "Iterative and sequential Kalman filter-based speech enhancement algorithms," *IEEE Transactions on Signal Processing*, vol. 6, no. 4, pp. 373–385, 1998.

[33] Garofolo, J. S., Lamel, L. F., Fisher, W. M., Fiscus, J. G., Pallett, D. S., and Dahlgren, N. L., "The DARPA TIMIT acoustic-phonetic continuous speech corpus," Feb. 1993. CDROM: NTIS order number PB91-100354.

[34] George, E. B., McCree, A. V., and Viswanathan, V. R., "Variable frame rate parameter encoding via adaptive frame selection using dynamic programming," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 271–274, 1996.

[35] Gersho, A. and Gray, R. M., *Vector Quantization and Signal Compression.* Kluwer Academic Publishers, 1991.

[36] Gersho, A. and Paskoy, E., "An overview of variable rate speech coding for cellular networks," in *IEEE Conference on Selected Topics in Wireless Communications*, pp. 172–175, 1992.

[37] Gibson, J., "Adaptive prediction in speech differential encoding systems," *Proceedings of the IEEE*, vol. 68, pp. 1789–1797, 1974.

[38] Gibson, J., "Backward adaptive prediction in multitree speech coders in *advances in speech coding*: B. Atal, V. Cuperman and A. Gersho," pp. 5–12, 1990.

[39] Gibson, J. D., Koo, B., and Gray, S. D., "Filtering of colored noise for speech enhancement and coding," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 39, pp. 1732–1742, 1991.

[40] Griffin, D. and Lim, J., "Multiband excitation vocoder," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, pp. 1223–1235, Oct. 1988.

[41] Gustafsson, H., Claesson, I., and Lindgren, U., "Speech bandwidth extension," in *Proceedings of the IEEE ICME Conference*, pp. 1016–1019, Aug. 2001.

[42] Hagen, R. and Hedelin, P., "Robust vector quantization by a linear mapping of block code," *IEEE Transactions on Information Theory*, vol. 45, no. 1, pp. 200–218, 1999.

[43] Hansen, J. H. L. and Clements, M. A., "Constrained iterative speech enhancement with application to speech recognition," *IEEE Transactions on Signal Processing*, vol. 39, pp. 795–805, 1991.

[44] Hayes, M. H., *Statistical Digital Signal Processing and Modeling*. John Wiley & Sons, Inc, 1996.

[45] Itakura, F., "Line spectrum representation of linear predictive coefficients of speech signals," *Journal of the Acoustical Society of America*, vol. 57, p. 535(a), 1975.

[46] ITU-T Recommendation G.712, "Transmission performance characteristics of pulse code modulation channels," ITU-T Rec.G712, International Telecommunications Union, Geneva, Switzerland, 1996.

[47] ITU-T Recommendation G726, "40, 32, 24, 16 kbit/s adaptive differential pulse code modulation (ADPCM)," ITU-T Rec.G726 Document Number E 1951, International Telecommunications Union, Geneva, Switzerland, 1991.

[48] ITU-T Recommendation G729, "Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear-prediction (CS-ACELP)," ITU-T Rec.G729 Document Number E 10204, International Telecommunications Union, Geneva, Switzerland, 1996.

[49] Juang, B.-H. and Gray, A. H., "Multiple stage vector quantization for speech coding," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 597–600, 1982.

[50] Juang, B.-H., Wong, D. Y., and Gray, A. H., "Distortion performance of vector quantization for LPC voice coding," *IEEE Transactions on Communications*, vol. 41, pp. 186–199, 1982.

[51] Kalman, R. E., "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, no. Series D, pp. 35–45, 1960.

[52] Kang, G. and Fransen, L., "Low–bit–rate speech coders based on line spectral frequencies LSFs," Naval Research Lab report no. 8857, 1984.

[53] Krishnan, V. and Anderson, D. V., "Robust jointly optimized multistage vector quantization for speech coding," in *Proceedings of Eurospeech*, pp. 1093–1096, 2003.

[54] Krishnan, V. and Anderson, D. V., "Joint design of channel-optimized multistage vector quantizers," *IEEE Signal Processing Letters*, vol. 11, no. 1, pp. 5–7, 2004.

[55] Krishnan, V., Anderson, D. V., and Truong, K. K., "Optimal multistage vector quantization of LPC parameters over noisy channel," *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 1, pp. 1–8, 2004.

[56] Kroon, P., Deprettere, E., and Sluyter, R., "Regular–pulse excitation-A novel approach to effective and efficient multipulse coding of speech," *IEEE Transactions on Speech and Audio Processing*, vol. 34, pp. 642–650, 1988.

[57] LeBlanc, W. P., Bhattacharya, B., Mahmoud, S. A., and Cuperman, V., "Efficient search and design procedures for robust multi–stage VQ of LPC parameters for 4 kb/s speech coding," *IEEE Transactions on Speech and Audio Processing*, vol. 1, pp. 373–385, Oct. 1993.

[58] Lee, B. G., Lee, K. Y., and Ann, S., "An EM based approach for parameter enhancement with application to speech signal," *Signal Processing*, vol. 46, pp. 1–14, 1995.

[59] Lee, K.-S. and Cox, R. V., "A very low bit–rate speech coder based on a recognition synthesis paradigm," *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 5, pp. 482–491, 2001.

[60] Lee, M. E., Durey, A. S., Moore, E., and Clements, M. A., "Ultra low bit rate speech coding using an ergodic hidden Markov model," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, (Philadelphia, PA), April 2005.

[61] Leonard, R. G., "A database for speaker independent digit recognition," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, (San Diego, CA), 1984.

[62] Leung, C. S. and Chen, L. W., "Transmission of vector quantized data over a noisy channel," *IEEE Transactions on Neural Networks*, vol. 8, no. 3, pp. 682–589, 1997.

[63] Lim, J. S. and Oppenheim, A. V., "All pole modelling of degraded speech," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, pp. 197–210, 1978.

[64] Lim, J. S. and Oppenheim, A. V., "Enhancement and bandwidth compression of noisy speech," *Proceedings of the IEEE*, vol. 67, pp. 1586–1604, 1979.

[65] Linden, J., "Channel-optimized predictive vector quantization," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 4, pp. 370–384, 2000.

[66] Lookabaugh, T., Riskin, E. A., Chou, P. A., and Gray, R. M., "Variable rate vector quantization for speech, image and video compression," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-30, pp. 294–304, 1993.

[67] Makhoul, J., "Linear prediction: A tutorial review," *Proceedings of the IEEE*, vol. 63, no. 4, pp. 561–580, 1975.

[68] Makhoul, J. and Berouti, M., "High-frequency regeneration in speech coding systems," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, pp. 428–431, Apr. 1979.

[69] Mammen, C. P. and Ramamurthi, B., "Capacity enhancement in digital cellular systems using variable bitrate speech coding," in *IEEE International Conference on Communication*, vol. 2, pp. 735–739, 1997.

[70] Manxia, T., Dusheng, W., and Changxin, F., "A novel variable-rate MELP speech coder," in *International Conference on Signal Processing*, vol. 2, pp. 21–25, 2000.

[71] Martin, R. and Cox, R., "New speech enhancement techniques for low bit-rate speech coding," in *Proceedings of the Speech Coding Workshop*, pp. 165–167, 1999.

[72] McAulay, R. and Quatieri, T., "Speech analysis/ synthesis based on sinusoidal representation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, pp. 744–754, 1986.

[73] McCree, A. V. and Barnwell-III, T. P., "A mixed excitation LPC vocoder model for low bit–rate speech coding," *IEEE Transactions on Speech and Audio Processing*, vol. 3, pp. 242–250, 1995.

[74] Paliwal, K. K. and Atal, B. S., "Efficient vector quantization of LPC parameters at 24 bits/frame," in *Proceedings of ICASSP*, pp. 661–664, 1991.

[75] Paliwal, K. K. and Basu, A., "A speech enhancement method based on Kalman filtering," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 177–180, 1987.

[76] Paskoy, E., Srinivasan, K., and Gersho, A., "Variable rate speech coding with phonetic segmentation," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. II–155–158, 1993.

[77] Pearce, D. and Hirsch, H. G., "The aurora experimental framework for the performance evaluatation of speech recognition system under noisy conditions," in *Proceedings of the 6th International Conference on Spoken Language Processing*, vol. 4, pp. 29–32, 2000.

[78] Phamdo, N., Farvardin, N., and Moriya, T., "A unified approach to tree structured and multistage vector for noisy channels," *IEEE Transactions on Information Theory*, vol. 39, pp. 835–850, May 1993.

[79] Quackenbush, S. R., Barnwell, T. P., and Clements, M. A., *Objective measures of speech quality*. Englewood Cliffs, NJ: Prentice Hall, 1988.

[80] Quatieri, T. F., *Discrete time speech signal processing*. Englewood Cliffs: Prentice-Hall, 2001.

[81] Rabiner, L. R., "A tutorial on Hidden Markov Models and selected applications in speech recognition,"

[82] Sangamura, N. and Itakura, T., "Speech data compression by LSP speech analysis and synthesis technique," *IECE Transactions*, vol. J64A, no. 8, pp. 599–605, 1981.

[83] Sayood, K., *Introduction to data compression*. Burlington: Morgan Kauffman, 2nd ed., 2000.

[84] Sheskin, D. J., *Handbook of parametric and non-parametric statistical procedures*. Boca Raton, FL: CRC Press, 1997.

[85] Soong, F. K. and Juang, B. H., "Line specrtal pair and speech data compression," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 1.10.1–1.10.4, 1984.

[86] Spanias, A. S., "Speech coding: A tutorial review," *Proceedings of the IEEE*, vol. 82, pp. 1541–1582, Oct. 1994.

[87] Street, M., "Future NATO narrow band voice coder selection (phase one): Stanag 4591," NC3A Technical Note 881, Den Haag, The Netherlands, 2002.

[88] Taubman, D. S. and Marcellin, M. W., *JPEG 2000: Image compression fundamentals, standards and practice*. Kluwer international series in engineering and computer science.

[89] Toledano, D. T., Gomez, L. A. H., and Grande, L. V., "Automatic phonetic segmentation," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, pp. 617–625, 2003.

[90] Tremain, T., "The government standard linear predictive coding algorithm: LPC-10," *Speech Technology*, pp. 40–49, 1980.

[91] Tugnait, J. K., "Adaptive estimation and identification for discrete systems with markov jump parameters," *IEEE Transactions on Automatic Control*, vol. 27, no. 5, pp. 1054–1065, 1981.

[92] Unno, T., *An Improved Mixed Excitation Linear Predictive (MELP) Coder*. Masters thesis, Georgia Institute of Technology, 1998.

[93] Varga, A. P., Steeneken, H. J. M., Tomlinson, M., and Jones, D., "The NOISEX-92 study on the effect of additive noise on ASR systems." Technical report, DRA speech research unit, Malvern, UK 1992.

[94] Vaseghi, S. V., "Finite state CELP for variable rate speech coding," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 37–40, 1990.

[95] Virag, N., "Single channel speech enhancement based on masking properties of the human auditory system," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 2, pp. 126–137, 1999.

[96] Wang, S. and Gersho, A., "Improved phonetically segmented vector excitation coding at 3.4 kb/s," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. I–349–352, 1992.

[97] Wang, T., Koishida, K., Cuperman, V., Gersho, A., and Collura, J., "A 1200 bps speech coder based on MELP," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 1375–1378, 2000.

[98] Wang, T., Koishida, K., Cuperman, V., Gersho, A., and Collura, J., "A 1200/ 2400 bps coding suite based on MELP," in *Proceedings of the Speech Coding Workshop*, pp. 90–92, 2002.

[99] Weinstein, E., Oppenheim, A. V., and Feder, M., "Signal enhancement using single and multisensor measurements," RLE Technical Report Rep 560, Massachussets Institute of Technology, Cambridge, MA, 1990.

[100] Whitehead, P. S., Anderson, D. V., and Clements, M. A., "Adaptive, acoustic noise suppression for speech enhancement," in *Proceedings of the International Conference on Multimedia and Exposition*, vol. 1, pp. 565–568, 2003.

[101] Wong, D., Juang, B.-H., and Gray, A. H., "An 800 bit/s vector quantization LPC vocoder," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 30, pp. 770–780, Oct. 1982.

[102] Zheng, Y. and Hasegawa-Johnson, M., "Acoustic segmentation using switching state Kalman filter," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 752–755, 2003.

VITA

Venkatesh Krishnan was born on October 25, 1976 in Pune, India. He received his Bachelor of Engineering degree in Electronics and Communications Engineering from the Birla Institute of Technology, Ranchi, India in 1999 and Master of Science in Electrical Engineering from the University of Central Florida, Orlando, FL in 2001. In the fall of 2001, he joined Prof. David Anderson's research group at the Center for Signal and Image Processing, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, as a PhD candidate and a Graduate Research Assistant. During this time, he also worked closely with Dr. Tom Barnwell, Dr. Kwan Truong and Dr. Mark Clements. After his graduation in May 2005, he plans to join Qualcomm's CDMA Technolgy (QCT) group as a Senior Engineer.

Venkatesh's research interest lies in the broad area of Digital Signal Processing, including robust coding of speech signals, implementation of DSP system on reconfigurable computing platforms, and real time signal processing applications. His research has resulted in 4 US patent disclosures, and 23 publications in refereed international journals and conferences. In December 2004, he was awarded the Center for Signal and Image Processing Outstanding Research award.